C8051F52x-53x

8/4/2 kB ISP Flash MCU Family

Analog Peripherals
- **12-Bit ADC**
  - ±1 LSB INL (C8051F52x/C8051F53x); no missing codes
  - Programmable throughput up to 200 ksps
  - Up to 6/16 external inputs
  - Data dependent windowed interrupt generator
  - Built-in temperature sensor
- **Comparator**
  - Programmable hysteresis and response time
  - Configurable as wake-up or reset source
  - Low current
- **POR/Brownout Detector**
- **Voltage Reference—1.5 to 2.2 V (programmable)**

On-Chip Debug
- On-chip debug circuitry facilitates full-speed, non-intrusive in-system debug (No emulator required)
- Provides breakpoints, single stepping
- Inspect/modify memory and registers
- Complete development kit

Supply Voltage 2.7 to 5.25 V
- Built-in LDO regulator

High Speed 8051 µC Core
- Pipelined instruction architecture; executes 70% of instructions in 1 or 2 system clocks
- Up to **25 MIPS** throughput with 25 MHz system clock
- Expanded interrupt handler

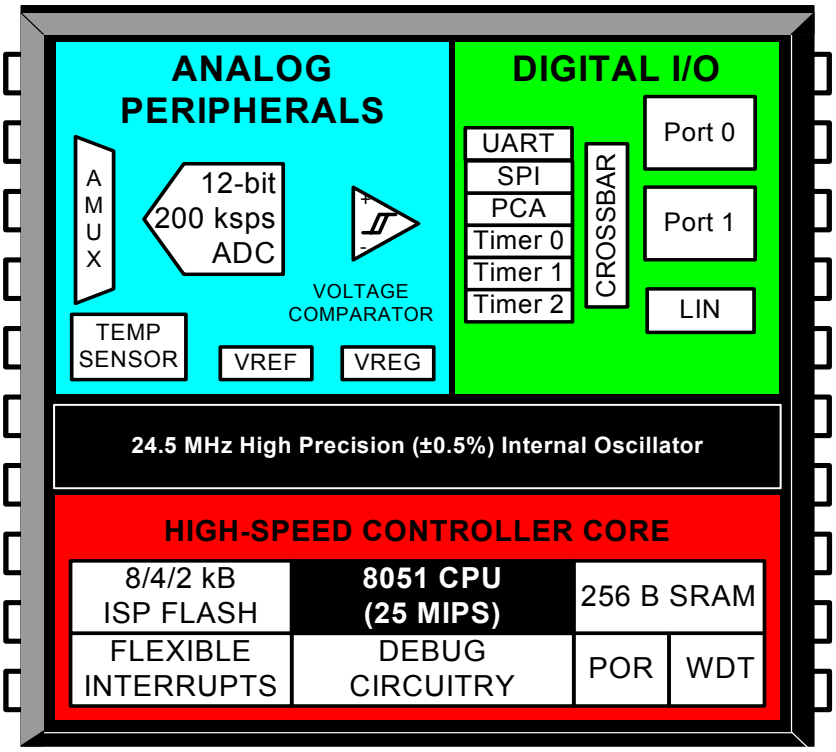
Memory
- 8/4/2 kB Flash; In-system byte programmable in 512 byte sectors
- 256 bytes internal data RAM

Digital Peripherals
- 16/6 port I/O; push-pull or open-drain, 5 V tolerant
- Hardware SPI™, and UART serial port
- Hardware LIN (both master and slave, compatible with V1.3 and V2.0)
- Three general purpose 16-bit counter/timers
- Programmable 16-bit counter/timer array with three capture/compare modules, WDT

Clock Sources
- Internal oscillators: 24.5 MHz **±0.5%** accuracy supports UART and LIN-Master operation
- External oscillator: Crystal, RC, C, or Clock (1 or 2 pin modes)
- Can switch between clock sources on-the-fly

Packages:
- **10-Pin QFN (3 x 3 mm)**
- **20-pin QFN (4 x 4 mm)**
- **20-pin TSSOP**

Temperature Range: –40 to +125 °C



NOTES:

Table of Contents

List of Figures

NOTES:

List of Tables

List of Registers

NOTES:

1. System Overview

The C8051F52x/C8051F53x family of devices are fully integrated, very low power, mixed-signal system-on-a-chip MCUs. Highlighted features are listed below. Refer to Table 1.1 for specific product feature selection.

- High-speed pipelined 8051-compatible microcontroller core (up to 25 MIPS)
- In-system, full-speed, non-intrusive debug interface (on-chip)
- True 12-bit 200 ksps ADC with analog multiplexer and up to 16 analog inputs
- Precision programmable 24.5 MHz internal oscillator that is ±0.5% across voltage and temperature
- Up to 7680 bytes of on-chip Flash memory
- 256 bytes of on-chip RAM
- Enhanced UART, and SPI serial interfaces implemented in hardware
- LIN 2.0 peripheral (V2.0 and V1.3 compatible, master and slave modes)
- Three general-purpose 16-bit timers
- Programmable Counter/Timer Array (PCA) with three capture/compare modules and Watchdog Timer function
- On-chip Power-On Reset, V_{DD} Monitor, and Temperature Sensor
- On-chip Voltage Comparator
- Up to 16 Port I/O

With on-chip Power-On Reset, V_{DD} monitor, Watchdog Timer, and clock oscillator, the C8051F52x/F53x devices are truly standalone system-on-a-chip solutions. The Flash memory is byte writable and can be reprogrammed in-circuit, providing non-volatile data storage, and also allowing field upgrades of the 8051 firmware. User software has complete control of all peripherals, and may individually shut down any or all peripherals for power savings.

The on-chip Silicon Laboratories 2-Wire (C2) Development Interface allows non-intrusive (uses no on-chip resources), full speed, in-circuit debugging using the production MCU installed in the final application. This debug logic supports inspection and modification of memory and registers, setting breakpoints, single stepping, run and halt commands. All analog and digital peripherals are fully functional while debugging using C2. The two C2 interface pins can be shared with user functions, allowing in-system programming and debugging without occupying package pins.

Each device is specified for 2.7 to 5.25 V operation (supply voltage can be up to 5.25 V using on-chip regulator) over the automotive temperature range (–40 to +125 °C). The F52x is available in the QFN10 (3 x 3 mm) package. The F53x is available in the QFN20 (4 x 4 mm) or the TSSOP20 package.

Table 1.1. Product Selection Guide

Ordering Part Number	MIPS (Peak)	Flash Memory	RAM	Calibrated Internal 24.5 MHz Oscillator Tolerance	SPI	UART	Timers (16-bit)	Programmable 3 Channels Counter Array	Port I/Os	12-bit ADC ±1 LSB INL	LIN	Internal Voltage Reference	Temperature Sensor	Analog Comparator	Package
C8051F520-IM	25	8 kB	256	0.5%	✓	✓	3	✓	8	✓	✓	✓	✓	✓	QFN-10
C8051F521-IM	25	8 kB	256	0.5%	✓	✓	3	✓	8	✓	—	✓	✓	✓	QFN-10
C8051F523-IM	25	4 kB	256	0.5%	✓	✓	3	✓	8	✓	✓	✓	✓	✓	QFN-10
C8051F524-IM	25	4 kB	256	0.5%	✓	✓	3	✓	8	✓	—	✓	✓	✓	QFN-10
C8051F526-IM	25	2 kB	256	0.5%	✓	✓	3	✓	8	✓	✓	✓	✓	✓	QFN-10
C8051F527-IM	25	2 kB	256	0.5%	✓	✓	3	✓	8	✓	—	✓	✓	✓	QFN-10
C8051F530-IM	25	8 kB	256	0.5%	✓	✓	3	✓	16	✓	✓	✓	✓	✓	QFN-20
C8051F531-IM	25	8 kB	256	0.5%	✓	✓	3	✓	16	✓	—	✓	✓	✓	QFN-20
C8051F533-IM	25	4 kB	256	0.5%	✓	✓	3	✓	16	✓	✓	✓	✓	✓	QFN-20
C8051F534-IM	25	4 kB	256	0.5%	✓	✓	3	✓	16	✓	—	✓	✓	✓	QFN-20
C8051F536-IM	25	2 kB	256	0.5%	✓	✓	3	✓	16	✓	✓	✓	✓	✓	QFN-20
C8051F537-IM	25	2 kB	256	0.5%	✓	✓	3	✓	16	✓	—	✓	✓	✓	QFN-20
C8051F530-IT	25	8 kB	256	0.5%	✓	✓	3	✓	16	✓	✓	✓	✓	✓	TSSOP-20
C8051F531-IT	25	8 kB	256	0.5%	✓	✓	3	✓	16	✓	—	✓	✓	✓	TSSOP-20
C8051F533-IT	25	4 kB	256	0.5%	✓	✓	3	✓	16	✓	✓	✓	✓	✓	TSSOP-20
C8051F534-IT	25	4 kB	256	0.5%	✓	✓	3	✓	16	✓	—	✓	✓	✓	TSSOP-20
C8051F536-IT	25	2 kB	256	0.5%	✓	✓	3	✓	16	✓	✓	✓	✓	✓	TSSOP-20
C8051F537-IT	25	2 kB	256	0.5%	✓	✓	3	✓	16	✓	—	✓	✓	✓	TSSOP-20

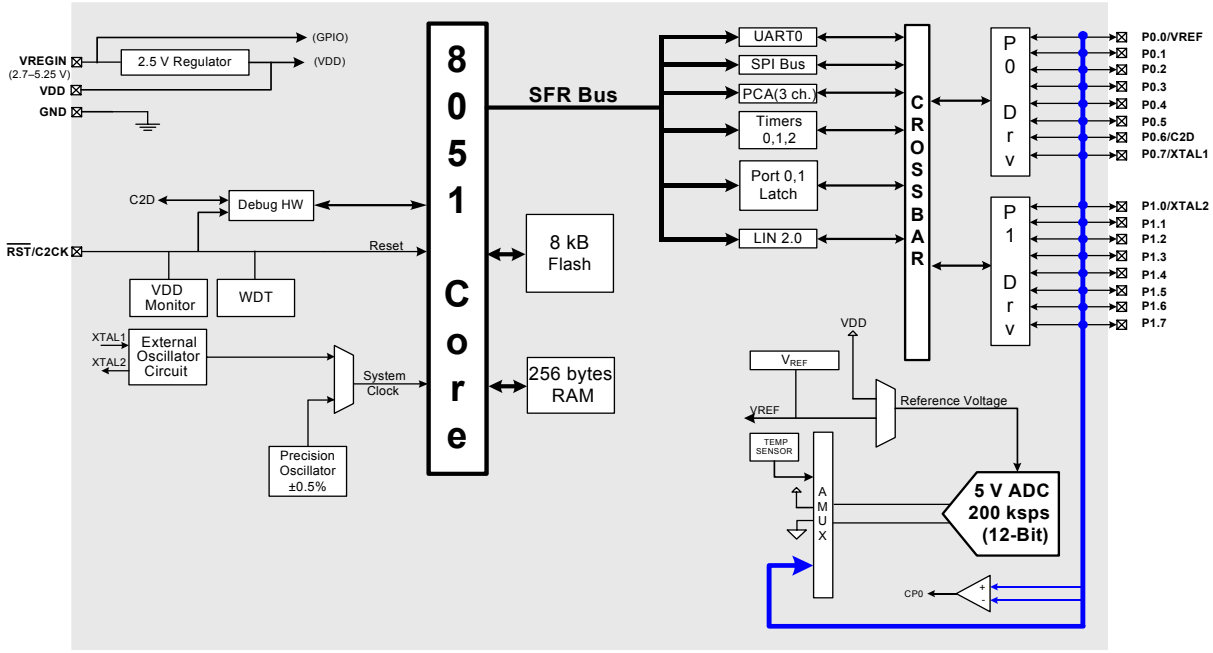


Figure 1.1. C8051F530 Block Diagram

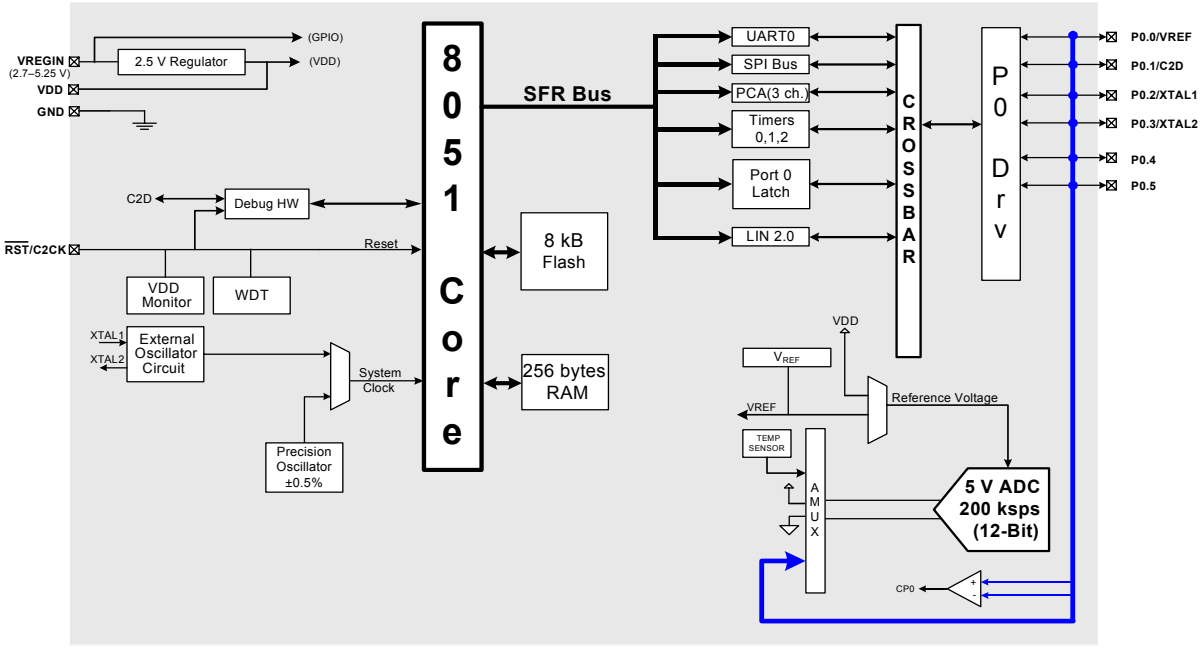


Figure 1.2. C8051F520 Block Diagram

1.1. CIP-51™ Microcontroller

1.1.1. Fully 8051 Compatible Instruction Set

The C8051F52x/F53xdevices use Silicon Laboratories' proprietary CIP-51 microcontroller core. The CIP-51 is fully compatible with the MCS-51™ instruction set. Standard 803x/805x assemblers and compilers can be used to develop software. The C8051F52x/F53xfamily has a superset of all the peripherals included with a standard 8052.

1.1.2. Improved Throughput

The CIP-51 employs a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. In a standard 8051, all instructions except for MUL and DIV take 12 or 24 system clock cycles to execute, and usually have a maximum system clock of 12-to-24 MHz. By contrast, the CIP-51 core executes 70% of its instructions in one or two system clock cycles, with no instructions taking more than eight system clock cycles.

With the CIP-51's system clock running at 25 MHz, it has a peak throughput of 25 MIPS. The CIP-51 has a total of 109 instructions. The table below shows the total number of instructions that require each execution time.

Clocks to Execute	1	2	2/4	3	3/5	4	5	4/6	6	8
Number of Instructions	26	50	5	10	7	5	2	1	2	1

1.1.3. Additional Features

The C8051F52x/F53x family includes several key enhancements to the CIP-51 core and peripherals to improve performance and ease of use in end applications.

An extended interrupt handler allows the numerous analog and digital peripherals to operate independently of the controller core and interrupt the controller only when necessary. By requiring less intervention from the microcontroller core, an interrupt-driven system is more efficient and allows for easier implementation of multi-tasking, real-time systems.

Eight reset sources are available: power-on reset circuitry (POR), an on-chip V_{DD} monitor, a Watchdog Timer, a Missing Clock Detector, a voltage level detection from Comparator, a forced software reset, an external reset pin, and an illegal Flash access protection circuit. Each reset source except for the POR, Reset Input Pin, or Flash error may be disabled by the user in software. The WDT may be permanently enabled in software after a power-on reset during MCU initialization.

The internal oscillator is factory calibrated to 24.5 MHz ±0.5% across the entire operating temperature and voltage range. An external oscillator drive circuit is also included, allowing an external crystal, ceramic resonator, capacitor, RC, or CMOS clock source to generate the system clock.

1.1.4. On-Chip Debug Circuitry

The C8051F52x/F53x devices include on-chip Silicon Laboratories 2-Wire (C2) debug circuitry that provides non-intrusive, full speed, in-circuit debugging of the production part *installed in the end application.*

Silicon Laboratories' debugging system supports inspection and modification of memory and registers, breakpoints, and single stepping. No additional target RAM, program memory, timers, or communications channels are required. All the digital and analog peripherals are functional and work correctly while debug-

ging. All the peripherals (except for the ADC) are stalled when the MCU is halted, during single stepping, or at a breakpoint in order to keep them synchronized.

The C8051F530-DK development kit provides all the hardware and software necessary to develop application code and perform in-circuit debugging with the C8051F52x/F53x MCUs. The kit includes software with a developer's studio and debugger, a USB debug adapter, a target application board with the associated MCU installed, and the required cables and wall-mount power supply. The development kit requires a computer with Windows installed. As shown in Figure 1.3, the PC is connected to the USB debug adapter. A six-inch ribbon cable connects the USB debug adapter to the user's application board, picking up the two C2 pins and GND.

The Silicon Laboratories IDE interface is a vastly superior developing and debugging configuration, compared to standard MCU emulators that use on-board "ICE Chips" and require the MCU in the application board to be socketed. Silicon Laboratories' debug paradigm increases ease of use and preserves the performance of the precision analog peripherals.

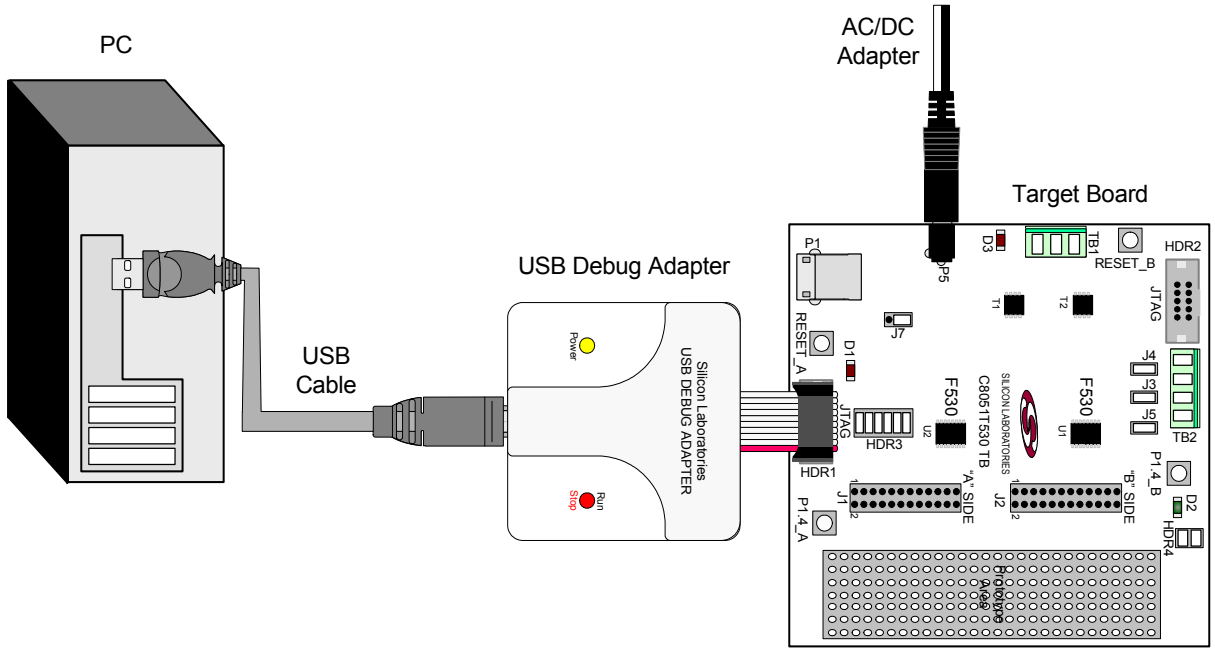


Figure 1.3. Development/In-System Debug Diagram

1.2. On-Chip Memory

The CIP-51 has a standard 8051 program and data address configuration. It includes 256 bytes of data RAM, with the upper 128 bytes dual-mapped. Indirect addressing accesses the upper 128 bytes of general purpose RAM, and direct addressing accesses the 128 byte SFR address space. The lower 128 bytes of RAM are accessible via direct and indirect addressing. The first 32 bytes are addressable as four banks of general purpose registers, and the next 16 bytes can be byte addressable or bit addressable.

Program memory consists of 7680 bytes ('F520/1 and 'F530/1), 4 kB ('F523/4 and 'F533/4), or 2 kB ('F526/7 and 'F536/7) of Flash. This memory is byte writable and erased in 512-byte sectors, and requires no special off-chip programming voltage.

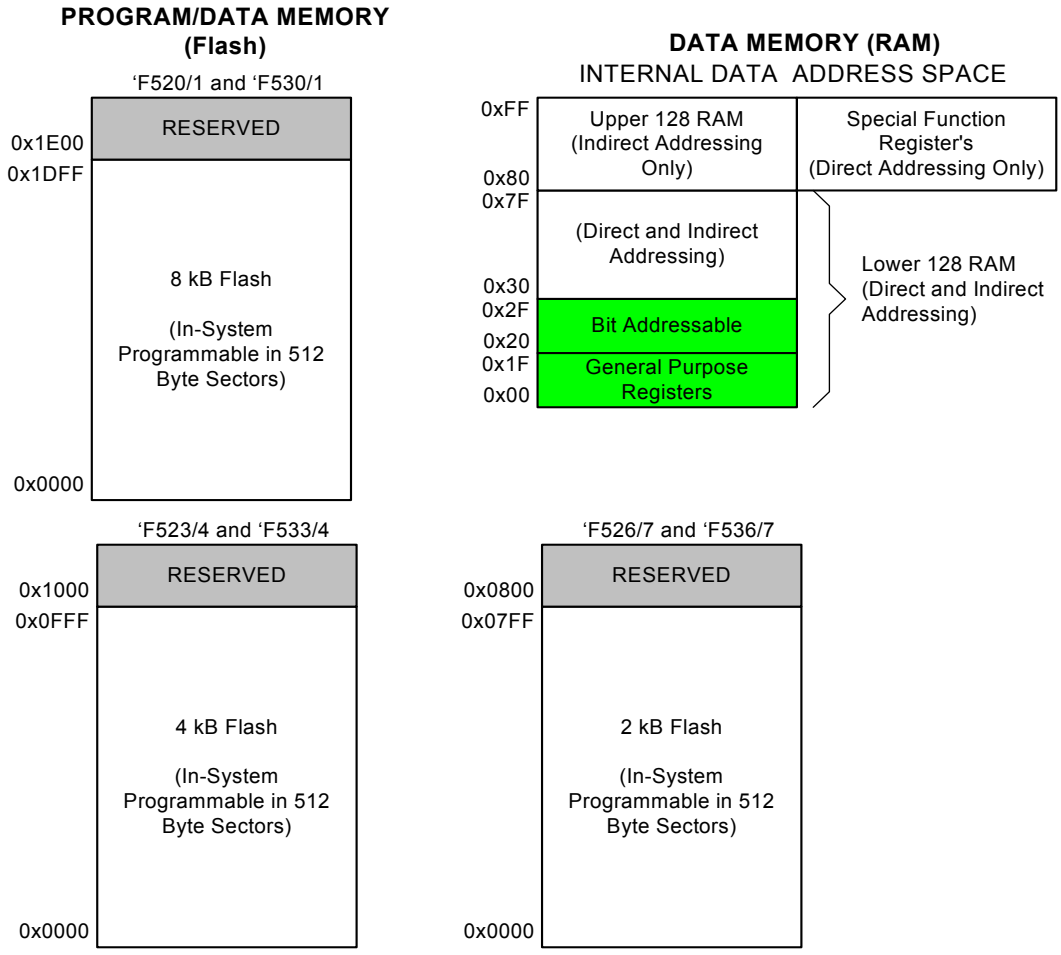


Figure 1.4. Memory Map

1.3. Operating Modes

The C8051F52x/F53x devices have four operating modes: Active (Normal), Idle, Suspend, and Stop. Active mode occurs during normal operation when the oscillator and peripherals are active. Idle mode halts the CPU while leaving the peripherals and internal clocks active. In Suspend and Stop mode, the CPU is halted, all interrupts and timers are inactive, and the internal oscillator is stopped. The various operating modes are described in Table 1.2 below:

Table 1.2. Operating Modes Summary

	Properties	Power Consumption	How Entered?	How Exited?
Active	• SYSCLK active • CPU active (accessing Flash) • Peripherals active or inactive depending on user settings	Full	—	—
Idle	• SYSCLK active • CPU inactive (not accessing Flash) • Peripherals active or inactive depending on user settings	Less than Full	IDLE (PCON.0)	Any enabled interrupt or device reset
Suspend	• Internal oscillator inactive • If SYSCLK is derived from the internal oscillator, the peripherals and the CIP-51 will be stopped	Low	SUSPEND (OSCICN.5)	Port 0 event match Port 1 event match Comparator 0 enabled and output is logic ‘0’
Stop	• SYSCLK inactive • CPU inactive (not accessing Flash) • Digital peripherals inactive; analog peripherals active or inactive depending on user settings	Very low	STOP (PCON.1)	Device Reset

See Section **“9.3. Power Management Modes” on page 81** for Idle and Stop mode details. See Section “**15.1.1. Internal Oscillator Suspend Mode**” on page **132** for more information on Suspend mode.

1.4. 12-Bit Analog to Digital Converter

The C8051F52x/F53x devices include an on-chip 12-bit SAR ADC with a maximum throughput of 200 ksps. The ADC system includes a configurable analog multiplexer that selects the positive ADC input, which is measured with respect to GND. Ports 0 and 1 are available as ADC inputs; additionally, the ADC includes an innovative half gain selection which allows for inputs up to twice the Vref voltage to be sampled. The on-chip Temperature Sensor output and the core supply voltage (V_{DD}) are also available as ADC inputs. User firmware may shut down the ADC or use it in Burst Mode to save power.

Conversions can be initiated in three ways: a software command, an overflow of Timer 2 or an external convert start signal. This flexibility allows the start of conversion to be triggered by software events, a periodic signal (timer overflows), or external HW signals. Conversion completions are indicated by a status bit and an interrupt (if enabled) and occur after 1, 4, 8, or 16 samples have been accumulated by a hardware accumulator. The resulting 12-bit to 16-bit data word is latched into the ADC data SFRs upon completion of a conversion. When the system clock is slow, Burst Mode allows ADC0 to automatically wake from a low power shutdown state, acquire and accumulate samples, then re-enter the low power shutdown state without CPU intervention.

Window compare registers for the ADC data can be configured to interrupt the controller when ADC data is either within or outside of a specified range. The ADC can monitor a key voltage continuously in background mode, but not interrupt the controller unless the converted data is within/outside the specified range.

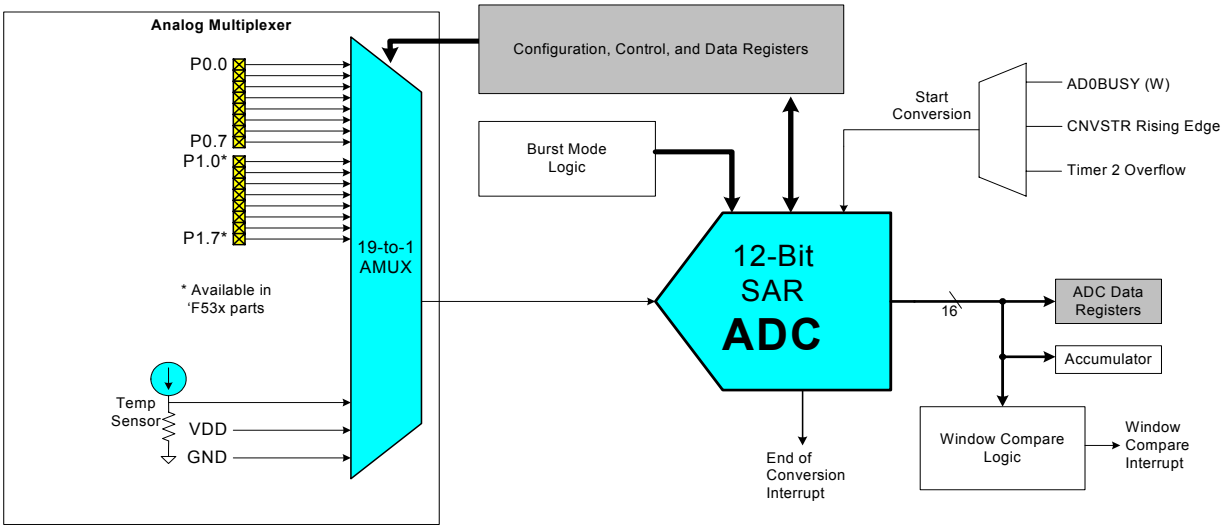


Figure 1.5. 12-Bit ADC Block Diagram

1.5. Programmable Comparator

C8051F52x/F53x devices include a software-configurable voltage comparator with an input multiplexer. The comparator offers programmable response time and hysteresis and an output that is optionally available at the Port pins: a synchronous “latched” output (CP0). The comparator interrupt may be generated on rising, falling, or both edges. When in IDLE or SUSPEND mode, these interrupts may be used as a “wake-up” source for the processor. The Comparator may also be configured as a reset source. A block diagram of the comparator is shown in Figure 1.6.

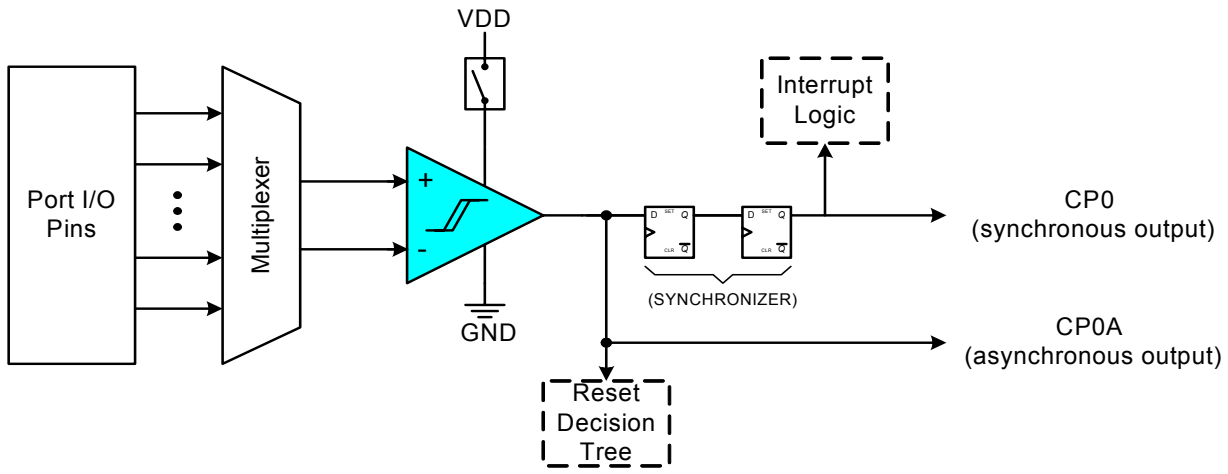


Figure 1.6. Comparator Block Diagram

1.6. Voltage Regulator

C8051F52x/F53x devices include an on-chip low dropout voltage regulator (REG0). The input to REG0 at the V_{REGIN} pin can be as high as 5.25 V. The output can be selected by software to 2.0 V or 2.5 V. When enabled, the output of REG0 powers the device and drives the V_{DD} pin. The voltage regulator can be used to power external devices connected to V_{DD}.

1.7. Serial Port

The C8051F52x/F53x Family includes a full-duplex UART with enhanced baud rate configuration, and an Enhanced SPI interface. Each of the serial buses is fully implemented in hardware and makes extensive use of the CIP-51's interrupts, thus requiring very little CPU intervention.

1.8. Port Input/Output

C8051F52x/F53x devices include up to 16 I/O pins. Port pins are organized as two byte-wide ports. The port pins behave like typical 8051 ports with a few enhancements. Each port pin can be configured as a digital or analog I/O pin. Pins selected as digital I/O can be configured for push-pull or open-drain operation. The “weak pullups” that are fixed on typical 8051 devices may be globally disabled to save power.

The Digital Crossbar allows mapping of internal digital system resources to port I/O pins. On-chip counter/timers, serial buses, hardware interrupts, and other digital signals can be configured to appear on the port pins using the Crossbar control registers. This allows the user to select the exact mix of general-purpose port I/O, digital, and analog resources needed for the application.

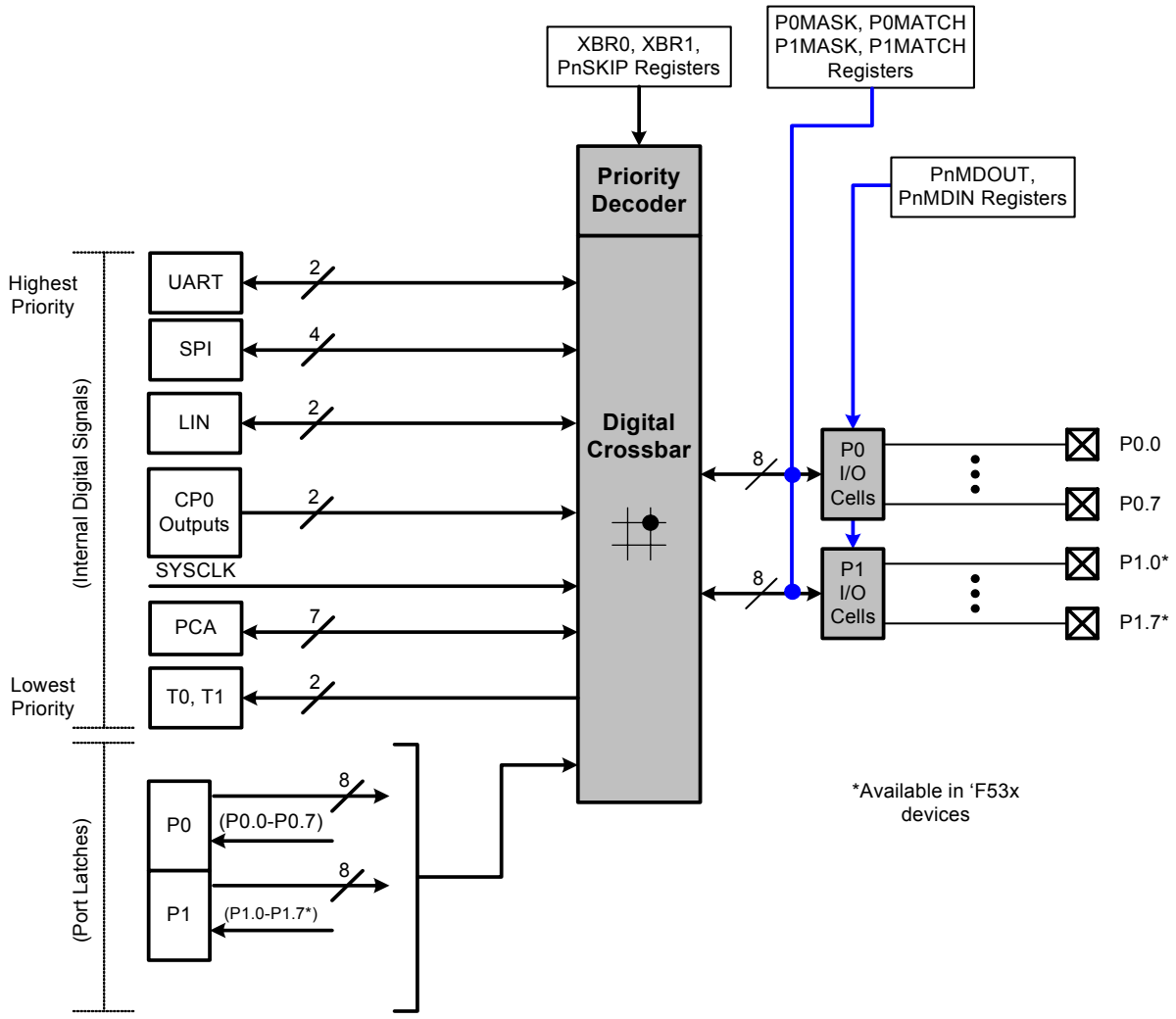


Figure 1.7. Port I/O Functional Block Diagram

NOTES:

2. Absolute Maximum Ratings

Table 2.1. Absolute Maximum Ratings

Parameter	Conditions	Min	Typ	Max	Units
Ambient temperature under bias		–55	—	125	°C
Storage Temperature		–65	—	150	°C
Voltage on V_{REGIN} with respect to GND		–0.3	—	5.5	V
Voltage on VDD with respect to GND		–0.3	—	2.8	V
Voltage on XTAL1 with respect to GND		–0.3	—	V_{DD} + 0.3	V
Voltage on XTAL2 with respect to GND		–0.3	—	V_{DD} + 0.3	V
Voltage on any Port I/O Pin (except Port 0 pins) or $\overline{RST}$ with respect to GND		–0.3	—	V_{IO} + 0.3	V
Voltage on any Port 0 Pin with respect to GND		0.3	—	5.5	V
Maximum output current sunk by any Port pin		—	—	100	mA
Maximum output current sourced by any Port pin		—	—	100	mA
Maximum Total current through V_{IO}, V_{REGIN}, and GND		—	—	500	mA

Note: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the devices at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

3. Global DC Electrical Characteristics

Table 3.1. Global DC Electrical Characteristics

–40 to +125 °C, 25 MHz System Clock unless otherwise specified. Typical values are given at 25 °C

Parameter	Conditions	Min.	Typ.	Max.	Units
Supply Input Voltage (V_{REGIN})[1]	Output Current = 1 mA	2.7	—	5.25	V
I/O Supply Voltage (V_{IO})		2.7	—	5.25	V
Core Supply Current with CPU active[2]	**V_{DD} = 2.0 V:**				
	Clock = 32 kHz	—	13	—	µA
	Clock = 200 kHz	—	40	—	µA
	Clock = 1 MHz	—	0.25	—	mA
	Clock = 25 MHz		9	—	mA
	V_{DD} = 2.4 V:	—			
	Clock = 32 kHz		21	—	µA
	Clock = 200 kHz	—	84	—	µA
	Clock = 1 MHz	—	0.45	—	mA
	Clock = 25 MHz	—	9	—	mA
Core Supply Current with CPU inactive (not accessing Flash)	**V_{DD} = 2.0 V:**				
	Clock = 32 kHz	—	10	—	µA
	Clock = 200 kHz	—	22	—	µA
	Clock = 1 MHz	—	0.15	—	mA
	Clock = 25 MHz	—	3	—	mA
	V_{DD} = 2.4 V:				
	Clock = 32 kHz	—	15	—	µA
	Clock = 200 kHz	—	34	—	µA
	Clock = 1 MHz	—	0.23	—	mA
	Clock = 25 MHz	—	4	—	mA
Core Supply Current (suspend)	Oscillator not running	—	TBD	—	µA
Core Supply Current (shutdown)	Oscillator not running	—	TBD	—	µA
Core Supply RAM Data Retention Voltage		—	TBD	—	V
SYSCLK (System Clock)[2]		0	—	25	MHz
Specified Operating Temperature Range		–40	—	+125	°C

Notes:

1. For more information on V_{REGIN} characteristics, see Table 7.1 on page 66.
2. SYSCLK must be at least 32 kHz to enable debugging.

4. Pinout and Package Definitions

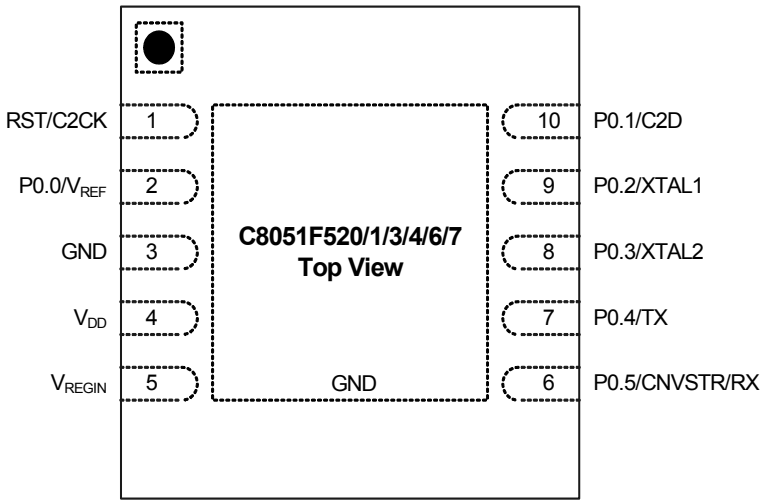


Table 4.1. Pin Definitions for the C8051F520 (QFN 10)

Name	Pin	Type	Description
$\overline{RST}$/	1	D I/O	Device Reset. Open-drain output of internal POR or V_{DD} monitor. An external source can initiate a system reset by driving this pin low for at least 15 µs. A 1 kΩ pullup to V_{DD} is recommended. See Reset Sources Section for a complete description.
C2CK		D I/O	Clock signal for the C2 Debug Interface.
P0.0/	2	D I/O or A In	Port 0.0. See Port I/O Section for a complete description.
V_{REF}		A O or D In	External V_{REF} Input. See V_{REF} Section.
GND	3		Ground.
V_{DD}	4		Core Supply Voltage.
V_{REGIN}	5		On-Chip Voltage Regulator Input.
P0.5/RX*/	6	D I/O or A In	Port 0.5. See Port I/O Section for a complete description.
CNVSTR		D In	External Converter start input for the ADC0, see **Section "5. 12-Bit ADC (ADC0)" on page 41** for a complete description.
P0.4/TX*	7	D I/O or A In	Port 0.4. See Port I/O Section for a complete description.
***Note:** Please refer to **Section "21. Revision Specific Behavior" on page 209**.			

Table 4.1. Pin Definitions for the C8051F520 (QFN 10) (Continued)

Name	Pin	Type	Description
P0.3		D I/O or A In	Port 0.3. See Port I/O Section for a complete description.
XTAL2	8	D I/O	External Clock Output. For an external crystal or resonator, this pin is the excitation driver. This pin is the external clock input for CMOS, capacitor, or RC oscillator configurations. See Section "15. Oscillators" on page 131.
P0.2		D I/O or	Port 0.2. See Port I/O Section for a complete description.
XTAL1	9	A In	External Clock Input. This pin is the external oscillator return for a crystal or resonator. Section "15. Oscillators" on page 131.
P0.1/		D I/O or A In	Port 0.1. See Port I/O Section for a complete description.
C2D	10	D I/O	Bi-directional data signal for the C2 Debug Interface
***Note:** Please refer to Section "21. Revision Specific Behavior" on page 209.			

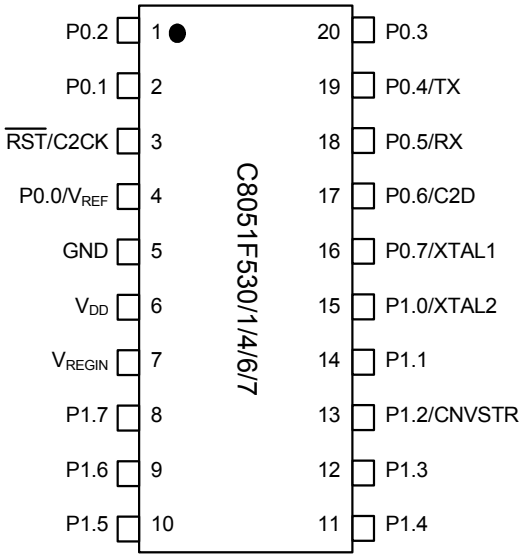


Table 4.2. Pin Definitions for the C8051F530 (TSSOP 20)

Name	Pin	Type	Description
P0.2	1	D I/O or A In	Port 0.2. See Port I/O Section for a complete description.
P0.1	2	D I/O or A In	Port 0.1. See Port I/O Section for a complete description.
$\overline{RST}$/	3	D I/O	Device Reset. Open-drain output of internal POR or V_{DD} monitor. An external source can initiate a system reset by driving this pin low for at least 15 µs. A 1 kΩ pullup to V_{DD} is recommended. See Reset Sources Section for a complete description.
C2CK		D I/O	Clock signal for the C2 Debug Interface.
P0.0/	4	D I/O or A In	Port 0.0. See Port I/O Section for a complete description.
V_{REF}		A O or D In	External V_{REF} Input. See V_{REF} Section.
GND	5		Ground.
V_{DD}	6		Core Supply Voltage.
V_{REGIN}	7		On-Chip Voltage Regulator Input.
P1.7	8	D I/O or A In	Port 1.7. See Port I/O Section for a complete description.
P1.6	9	D I/O or A In	Port 1.6. See Port I/O Section for a complete description.
***Note:** Please refer to **Section "21. Revision Specific Behavior" on page 209**.			

Table 4.2. Pin Definitions for the C8051F530 (TSSOP 20) (Continued)

Name	Pin	Type	Description
P1.5	10	D I/O or A In	Port 1.5. See Port I/O Section for a complete description.
P1.4	11	D I/O or A In	Port 1.4. See Port I/O Section for a complete description.
P1.3	12	D I/O or A In	Port 1.3. See Port I/O Section for a complete description.
P1.2/ CNVSTR	13	D I/O or A In D In	Port 1.2. See Port I/O Section for a complete description. External Converter start input for the ADC0, see **Section "5. 12-Bit ADC (ADC0)" on page 41** for a complete description.
P1.1	14	D I/O or A In	Port 1.1. See Port I/O Section for a complete description.
P1.0/ XTAL2	15	D I/O or A In D I/O	Port 1.0. See Port I/O Section for a complete description. External Clock Output. For an external crystal or resonator, this pin is the excitation driver. This pin is the external clock input for CMOS, capacitor, or RC oscillator configurations. See Section "**15. Oscillators**" on page **131**.
P0.7/ XTAL1	16	D I/O or A In A In	Port 0.7. See Port I/O Section for a complete description. External Clock Input. This pin is the external oscillator return for a crystal or resonator. Section "**15. Oscillators**" on page **131**.
P0.6/ C2D	17	D I/O or A In D I/O	Port 0.6. See Port I/O Section for a complete description. Bi-directional data signal for the C2 Debug Interface.
P0.5/RX*	18	D I/O or A In	Port 0.5. See Port I/O Section for a complete description.
P0.4/TX*	19	D I/O or A In	Port 0.4. See Port I/O Section for a complete description.
P0.3	20	D I/O or A In	Port 0.3. See Port I/O Section for a complete description.
***Note:** Please refer to **Section "21. Revision Specific Behavior" on page 209**.			

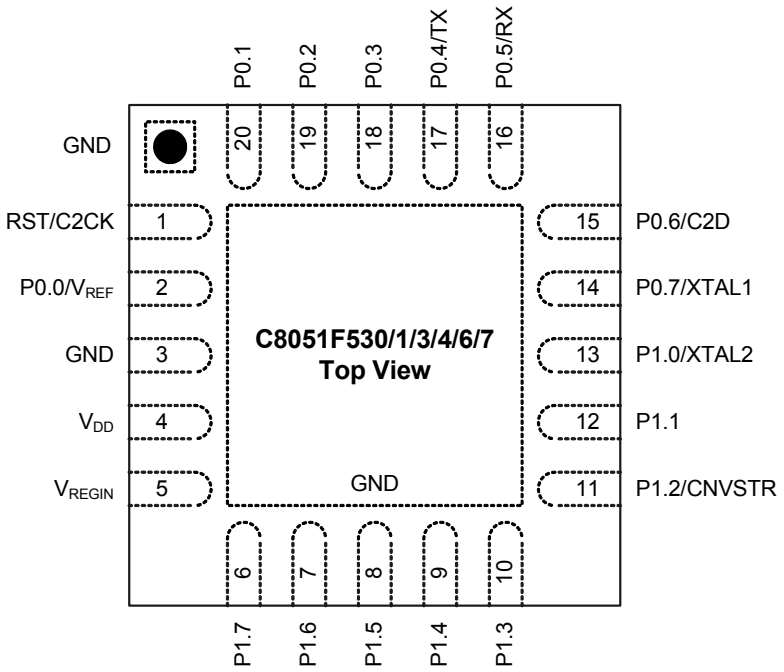


Table 4.3. Pin Definitions for the C8051F530 (QFN 20)

Name	Pin	Type	Description
$\overline{RST}$/	1	D I/O	Device Reset. Open-drain output of internal POR or V_{DD} monitor. An external source can initiate a system reset by driving this pin low for at least 15 µs. A 1 kΩ pullup to V_{DD} is recommended. See Reset Sources Section for a complete description.
C2CK		D I/O	Clock signal for the C2 Debug Interface.
P0.0/	2	D I/O or A In	Port 0.0. See Port I/O Section for a complete description.
V_{REF}		A O or D In	External V_{REF} Input. See V_{REF} Section.
GND	3		Ground.
V_{DD}	4		Core Supply Voltage.
V_{REGIN}	5		On-Chip Voltage Regulator Input.
P1.7	6	D I/O or A In	Port 1.7. See Port I/O Section for a complete description.
P1.6	7	D I/O or A In	Port 1.6. See Port I/O Section for a complete description.
P1.5	8	D I/O or A In	Port 1.5. See Port I/O Section for a complete description.
***Note:** Please refer to **Section "21. Revision Specific Behavior" on page 209**.			

Table 4.3. Pin Definitions for the C8051F530 (QFN 20) (Continued)

Name	Pin	Type	Description
P1.4	9	D I/O or A In	Port 1.4. See Port I/O Section for a complete description.
P1.3	10	D I/O or A In	Port 1.3. See Port I/O Section for a complete description.
P1.2/	11	D I/O or A In	Port 1.2. See Port I/O Section for a complete description.
CNVSTR		D In	External Converter start input for the ADC0, see **Section "5. 12-Bit ADC (ADC0)" on page 41** for a complete description.
P1.1	12	D I/O or A In	Port 1.1. See Port I/O Section for a complete description.
P1.0/	13	D I/O or A In	Port 1.0. See Port I/O Section for a complete description.
XTAL2		D I/O	External Clock Output. For an external crystal or resonator, this pin is the excitation driver. This pin is the external clock input for CMOS, capacitor, or RC oscillator configurations. Section "**15. Oscillators**" on page **131**.
P0.7/	14	D I/O or	Port 0.7. See Port I/O Section for a complete description.
XTAL1		A In	External Clock Input. This pin is the external oscillator return for a crystal or resonator. See Oscillator Section.
P0.6/	15	D I/O or A In	Port 0.6. See Port I/O Section for a complete description.
C2D		D I/O	Bi-directional data signal for the C2 Debug Interface.
P0.5/RX*	16	D I/O or A In	Port 0.5. See Port I/O Section for a complete description.
P0.4/TX*	17	D I/O or A In	Port 0.4. See Port I/O Section for a complete description.
P0.3	18	D I/O or A In	Port 0.3. See Port I/O Section for a complete description.
P0.2	19	D I/O or A In	Port 0.2. See Port I/O Section for a complete description.
P0.1	20	D I/O or A In	Port 0.1. See Port I/O Section for a complete description.
***Note:** Please refer to **Section "21. Revision Specific Behavior" on page 209**.			

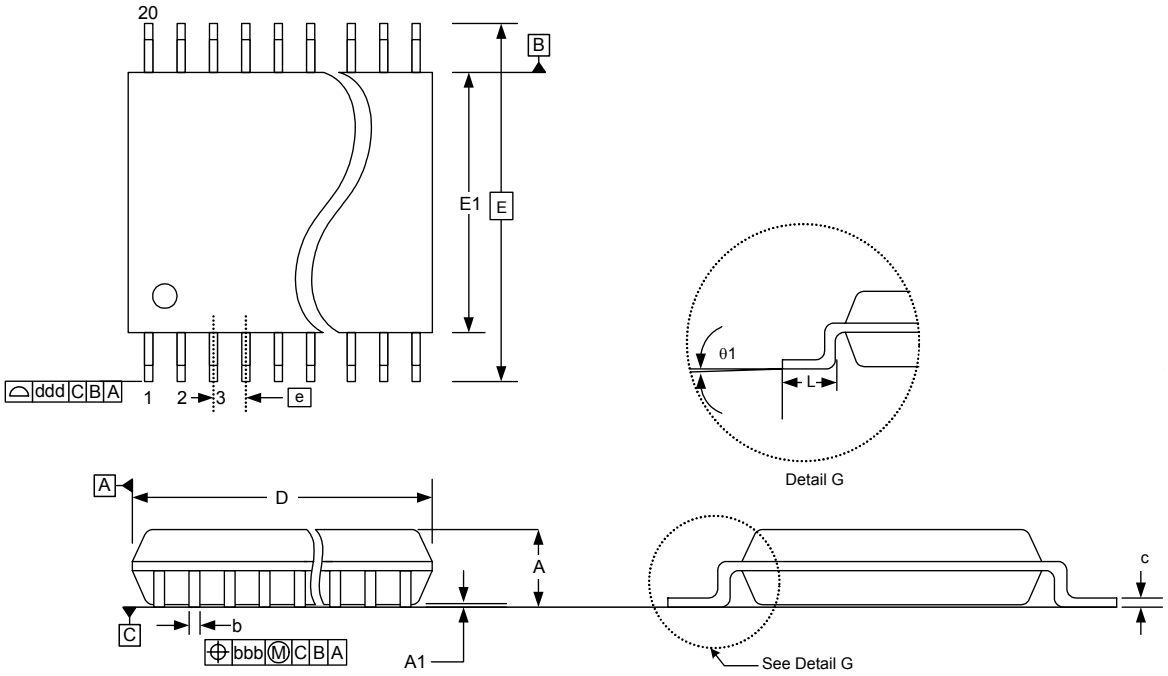


Figure 4.1. TSSOP-20 Package Diagram

Table 4.4. TSSOP-20 Package Diagram Dimensions

Symbol	Millimeters		
	Min	Nom	Max
A	—	—	1.20
A1	0.05	—	0.15
b	0.19	—	0.30
c	0.09	—	0.20
D	6.40	6.50	6.60
e	0.65 BSC		
E	6.40 BSC		
E1	4.30	4.40	4.50
L	0.45	0.60	0.75
θ1	0°	—	8°
bbb	0.10		
ddd	0.20		

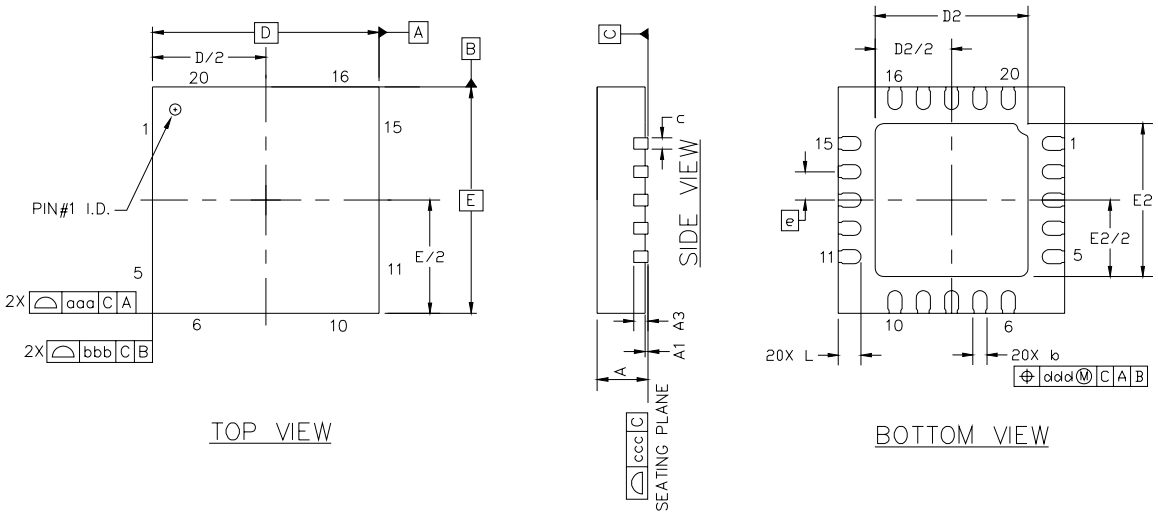


Figure 4.2. QFN-20 Package Diagram

Table 4.5. QFN-20 Package Diagram Dimensions

Dimension	MIN	NOM	MAX
A	0.80	0.90	1.00
A1	0.00	0.02	0.05
A3		0.25 TYP.	
b	0.20	0.25	0.30
c		0.204 TYP.	
D		4.00 BSC.	
D2	2.55	2.70	2.85
e		0.50 BSC.	
E		4.00 BSC.	
E2	2.55	2.70	2.85
L	0.30	0.40	0.50
aaa	--	--	0.10
bbb	--	--	0.10
ccc	--	--	0.08
ddd	--	--	0.10

Notes:

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. This drawing conforms to JEDEC outline MO-220, variation VGGD-5
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020C specification for Small Body Components.

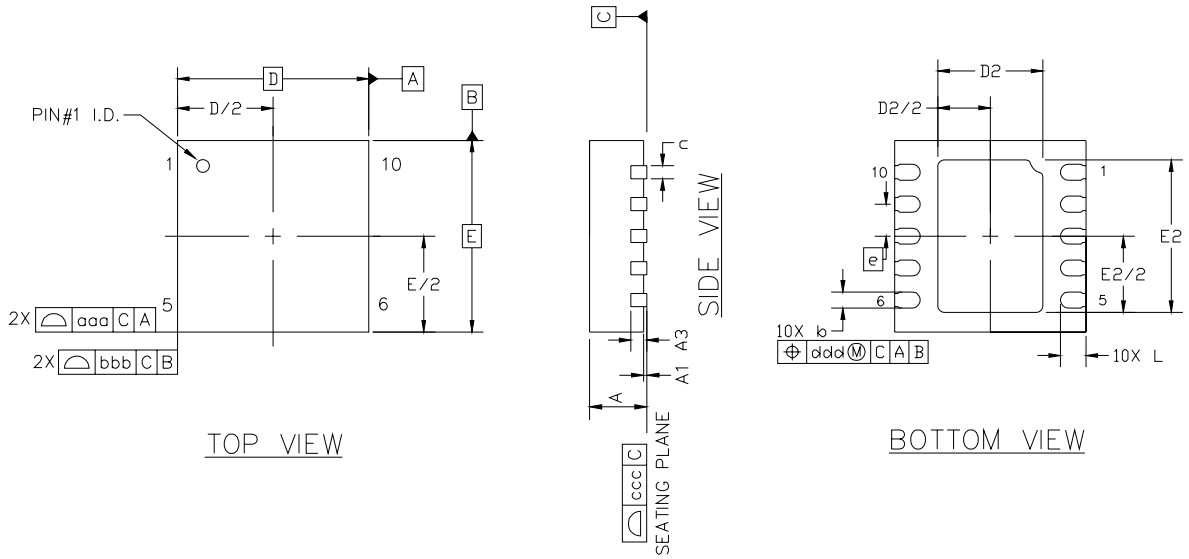


Figure 4.3. QFN-10 Package Diagram

Table 4.6. QFN-10 Package Diagram Dimensions

Dimension	MIN	NOM	MAX
A	0.80	0.90	1.00
A1	0.00	0.02	0.05
A3		0.25 TYP.	
b	0.20	0.25	0.30
c		0.204 TYP.	
D		3.00 BSC.	
D2	1.496	1.646	1.796
e		0.50 BSC.	
E		3.00 BSC.	
E2	2.234	2.384	2.534
L	0.30	0.40	0.50
aaa	--	--	0.10
bbb	--	--	0.10
ccc	--	--	0.08
ddd	--	--	0.10

Notes:

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. Recommended card reflow profile is per the JEDEC/IPC J-STD-020C specification for Small Body Components.

NOTES:

5. 12-Bit ADC (ADC0)

The ADC0 subsystem for the C8051F52x/F53x Family consists of an analog multiplexer (AMUX0) with 16/6 total input selections, and a 200 ksps, 12-bit successive-approximation-register ADC with integrated track-and-hold, programmable window detector, programmable attenuation (1:2), and hardware accumulator. The ADC0 subsystem has a special Burst Mode which can automatically enable ADC0, capture and accumulate samples, then place ADC0 in a low power shutdown mode without CPU intervention. The AMUX0, data conversion modes, and window detector are all configurable under software control via the Special Function Registers shown in Figure 5.1. ADC0 inputs are single-ended and may be configured to measure P0.0-P2.7, the Temperature Sensor output, V_{DD}, or GND with respect to GND. ADC0 is enabled when the AD0EN bit in the ADC0 Control register (ADC0CN) is set to logic 1, or when performing conversions in Burst Mode. ADC0 is in low power shutdown when AD0EN is logic 0 and no Burst Mode conversions are taking place.

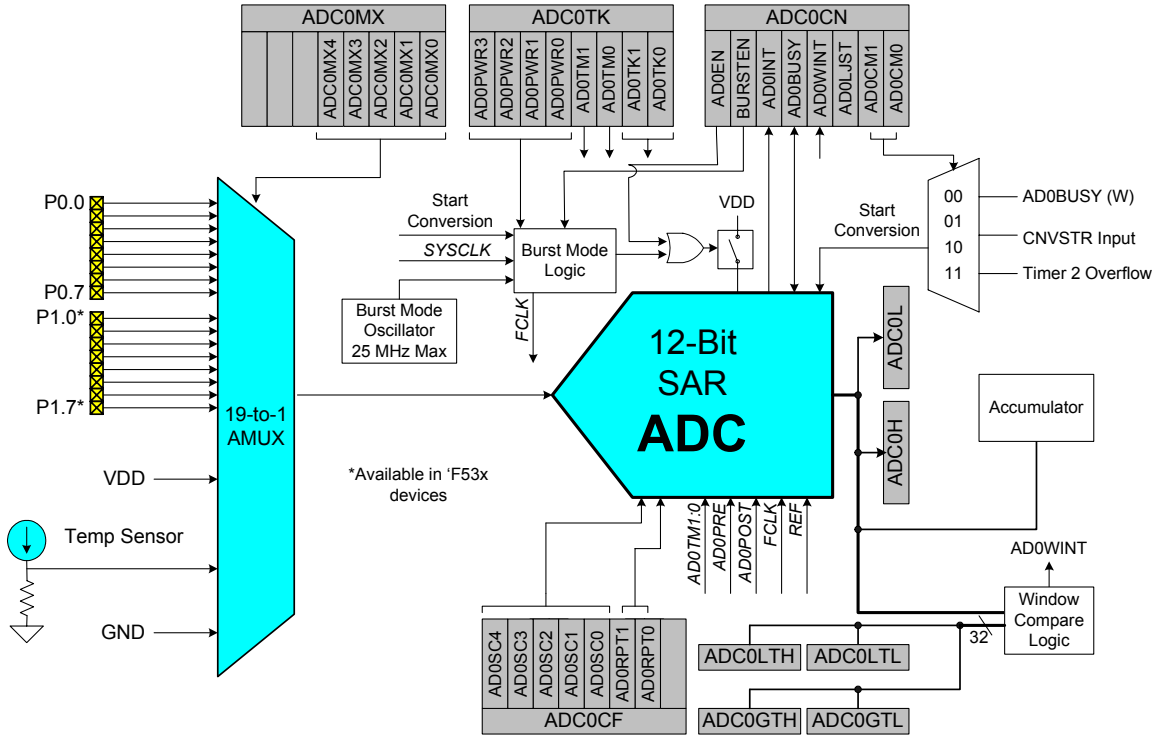


Figure 5.1. ADC0 Functional Block Diagram

5.1. Analog Multiplexer

AMUX0 selects the input channel to the ADC. Any of the following may be selected as an input: P0.0–P1.7, the on-chip temperature sensor, the core power supply (V_{DD}), or ground (GND). **ADC0 is single-ended and all signals measured are with respect to GND.** The ADC0 input channels are selected using the ADC0MX register as described in SFR Definition 5.1.

Important Note About ADC0 Input Configuration: Port pins selected as ADC0 inputs should be configured as analog inputs, and should be skipped by the Digital Crossbar. To configure a Port pin for analog input, set to '0' the corresponding bit in register PnMDIN (for n = 0,1). To force the Crossbar to skip a Port

pin, set to ‘1’ the corresponding bit in register PnSKIP (for n = 0,1). See **Section “14. Port Input/Output” on page 115** for more Port I/O configuration details.

5.2. Temperature Sensor

The typical temperature sensor transfer function is shown in Figure 5.2. The output voltage (V_{TEMP}) is the positive ADC input when the temperature sensor is selected by bits AD0MX4–0 in register ADC0MX.

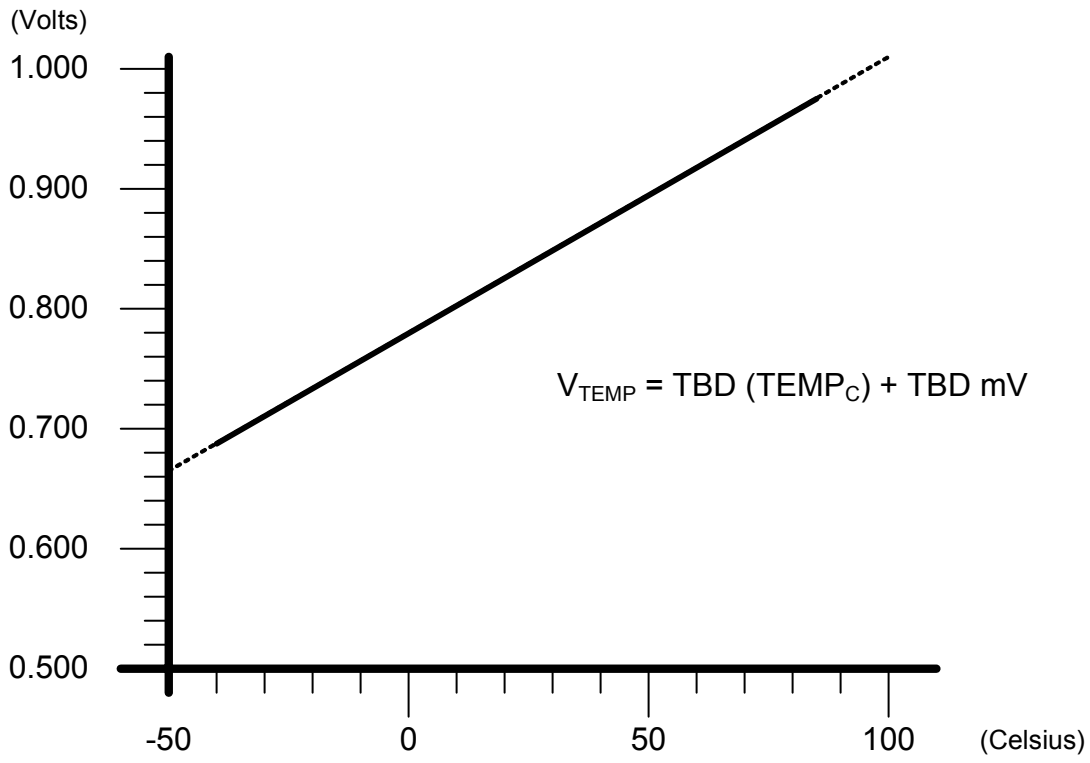


Figure 5.2. Typical Temperature Sensor Transfer Function

5.3. ADC0 Operation

In a typical system, ADC0 is configured using the following steps:

Step 1. If an attenuation (1:2) is required please refer to Section “**5.5. Selectable Attenuation**” on page **57**.
Step 2. Choose the start of conversion source.
Step 3. Choose Normal Mode or Burst Mode operation.
Step 4. If Burst Mode, choose the ADC0 Idle Power State and set the Power-Up Time.
Step 5. Choose the tracking mode. Note that Pre-Tracking Mode can only be used with Normal Mode.
Step 6. Calculate required settling time and set the post convert-start tracking time using the AD0TK bits.
Step 7. Choose the repeat count.
Step 8. Choose the output word justification (Right-Justified or Left-Justified).
Step 9. Enable or disable the End of Conversion and Window Comparator Interrupts.

5.3.1. Starting a Conversion

A conversion can be initiated in one of four ways, depending on the programmed states of the ADC0 Start of Conversion Mode bits (AD0CM1–0) in register ADC0CN. Conversions may be initiated by one of the following:

- Writing a ‘1’ to the AD0BUSY bit of register ADC0CN
- A rising edge on the CNVSTR input signal (pin P0.6)
- A Timer 2 overflow (i.e., timed continuous conversions)

Writing a ‘1’ to AD0BUSY provides software control of ADC0 whereby conversions are performed "on-demand.” During conversion, the AD0BUSY bit is set to logic 1 and reset to logic 0 when the conversion is complete. The falling edge of AD0BUSY triggers an interrupt (when enabled) and sets the ADC0 interrupt flag (AD0INT). Note: When polling for ADC conversion completions, the ADC0 interrupt flag (AD0INT) should be used. Converted data is available in the ADC0 data registers, ADC0H:ADC0L, when bit AD0INT is logic 1. Note that when Timer 2 overflows are used as the conversion source, Low Byte overflows are used if Timer2 is in 8-bit mode; High byte overflows are used if Timer2 is in 16-bit mode. See **Section “19. Timers” on page 179** for timer configuration.

5.3.2. Tracking Modes

According to Table 5.1 and Table 5.2, each ADC0 conversion must be preceded by a minimum tracking time for the converted result to be accurate. ADC0 has three tracking modes: Pre-Tracking, Post-Tracking, and Dual-Tracking. Pre-Tracking Mode provides the minimum delay between the convert start signal and end of conversion by tracking continuously before the convert start signal. This mode requires software management in order to meet minimum tracking requirements. In Post-Tracking Mode, a programmable tracking time starts after the convert start signal and is managed by hardware. Dual-Tracking Mode maximizes tracking time by tracking before and after the convert start signal. Figure 5.3 shows examples of the three tracking modes.

Pre-Tracking Mode is selected when AD0TM is set to 10b. Conversions are started immediately following the convert start signal. ADC0 is tracking continuously when not performing a conversion. Software must allow at least the minimum tracking time between each end of conversion and the next convert start signal. The minimum tracking time must also be met prior to the first convert start signal after ADC0 is enabled.

Post-Tracking Mode is selected when AD0TM is set to 01b. A programmable tracking time based on AD0TK is started immediately following the convert start signal. Conversions are started after the programmed tracking time ends. After a conversion is complete, ADC0 does not track the input. Rather, the sampling capacitor remains disconnected from the input making the input pin high-impedance until the next convert start signal.

Dual-Tracking Mode is selected when AD0TM is set to 11b. A programmable tracking time based on AD0TK is started immediately following the convert start signal. Conversions are started after the programmed tracking time ends. After a conversion is complete, ADC0 tracks continuously until the next conversion is started.

Depending on the output connected to the ADC input, additional tracking time, more than is specified in Table 5.1 and Table 5.2, may be required after changing MUX settings. See the settling time requirements described in **Section “5.3.6. Settling Time Requirements” on page 48**.

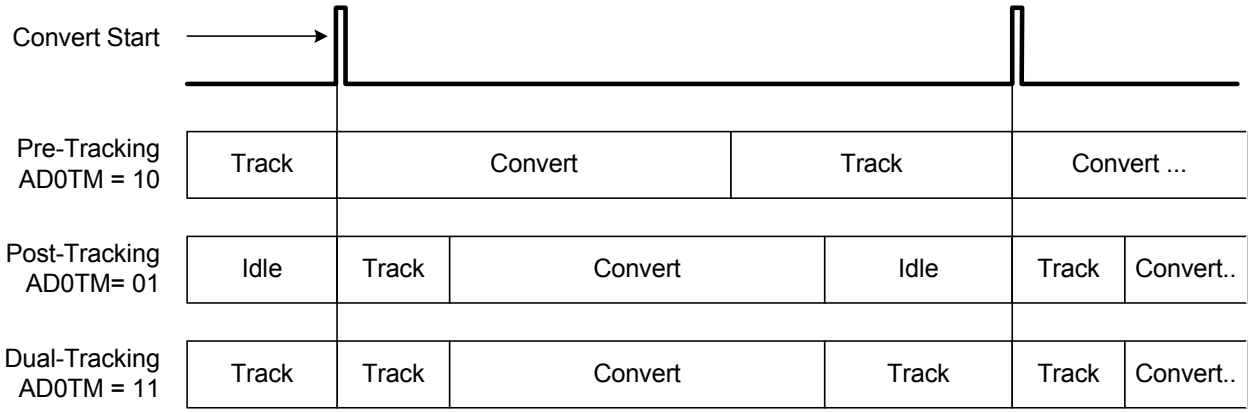


Figure 5.3. ADC0 Tracking Modes

5.3.3. Timing

ADC0 has a maximum conversion speed specified in Table 5.1 and Table 5.2. ADC0 is clocked from the ADC0 Subsystem Clock (FCLK). The source of FCLK is selected based on the BURSTEN bit. When BURSTEN is logic 0, FCLK is derived from the current system clock. When BURSTEN is logic 1, FCLK is derived from the Burst Mode Oscillator, an independent clock source with a maximum frequency of 25 MHz.

When ADC0 is performing a conversion, it requires a clock source that is typically slower than FCLK. The ADC0 SAR conversion clock (SAR clock) is a divided version of FCLK. The divide ratio can be configured using the AD0SC bits in the ADC0CF register. The maximum SAR clock frequency is listed in Table 5.1 and Table 5.2.

ADC0 can be in one of three states at any given time: tracking, converting, or idle. Tracking time depends on the tracking mode selected. For Pre-Tracking Mode, tracking is managed by software and ADC0 starts conversions immediately following the convert start signal. For Post-Tracking and Dual-Tracking Modes, the tracking time after the convert start signal is equal to the value determined by the AD0TK bits plus 2 FCLK cycles. Tracking is immediately followed by a conversion. The ADC0 conversion time is always 13 SAR clock cycles plus an additional 2 FCLK cycles to start and complete a conversion. Figure 5.4 shows timing diagrams for a conversion in Pre-Tracking Mode and tracking plus conversion in Post-Tracking or Dual-Tracking Mode. In this example, repeat count is set to one.

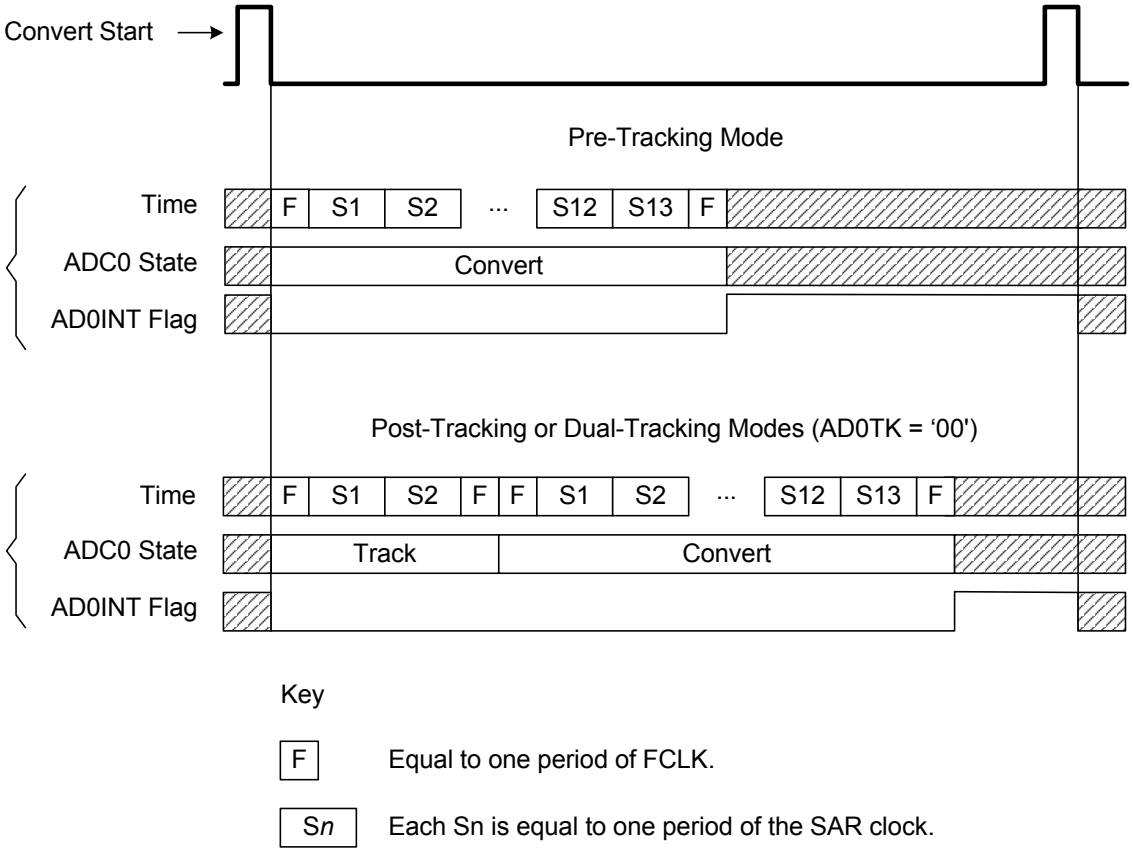


Figure 5.4. 12-Bit ADC Tracking Mode Example

5.3.4. Burst Mode

Burst Mode is a power saving feature that allows ADC0 to remain in a very low power state between conversions. When Burst Mode is enabled, ADC0 wakes from a very low power state, accumulates 1, 4, 8, or 16 samples using an internal Burst Mode clock (approximately 25 MHz), then re-enters a very low power state. Since the Burst Mode clock is independent of the system clock, ADC0 can perform multiple conversions then enter a very low power state within a single system clock cycle, even if the system clock is slow (e.g. 32.768 kHz), or suspended.

Burst Mode is enabled by setting BURSTEN to logic 1. When in Burst Mode, AD0EN controls the ADC0 idle power state (i.e. the state ADC0 enters when not tracking or performing conversions). If AD0EN is set to logic 0, ADC0 is powered down after each burst. If AD0EN is set to logic 1, ADC0 remains enabled after each burst. On each convert start signal, ADC0 is awakened from its Idle Power State. If ADC0 is powered down, it will automatically power up and wait the programmable Power-Up Time controlled by the AD0PWR bits. Otherwise, ADC0 will start tracking and converting immediately. Figure 5.5 shows an example of Burst Mode Operation with a slow system clock and a repeat count of 4.

Important Note: When Burst Mode is enabled, only Post-Tracking and Dual-Tracking modes can be used.

When Burst Mode is enabled, a single convert start will initiate a number of conversions equal to the repeat count. When Burst Mode is disabled, a convert start is required to initiate each conversion. In both modes, the ADC0 End of Conversion Interrupt Flag (AD0INT) will be set after “repeat count” conversions have been accumulated. Similarly, the Window Comparator will not compare the result to the greater-than and less-than registers until “repeat count” conversions have been accumulated.

Note: When using Burst Mode, care must be taken to issue a convert start signal no faster than once every four SYSCLK periods. This includes external convert start signals.

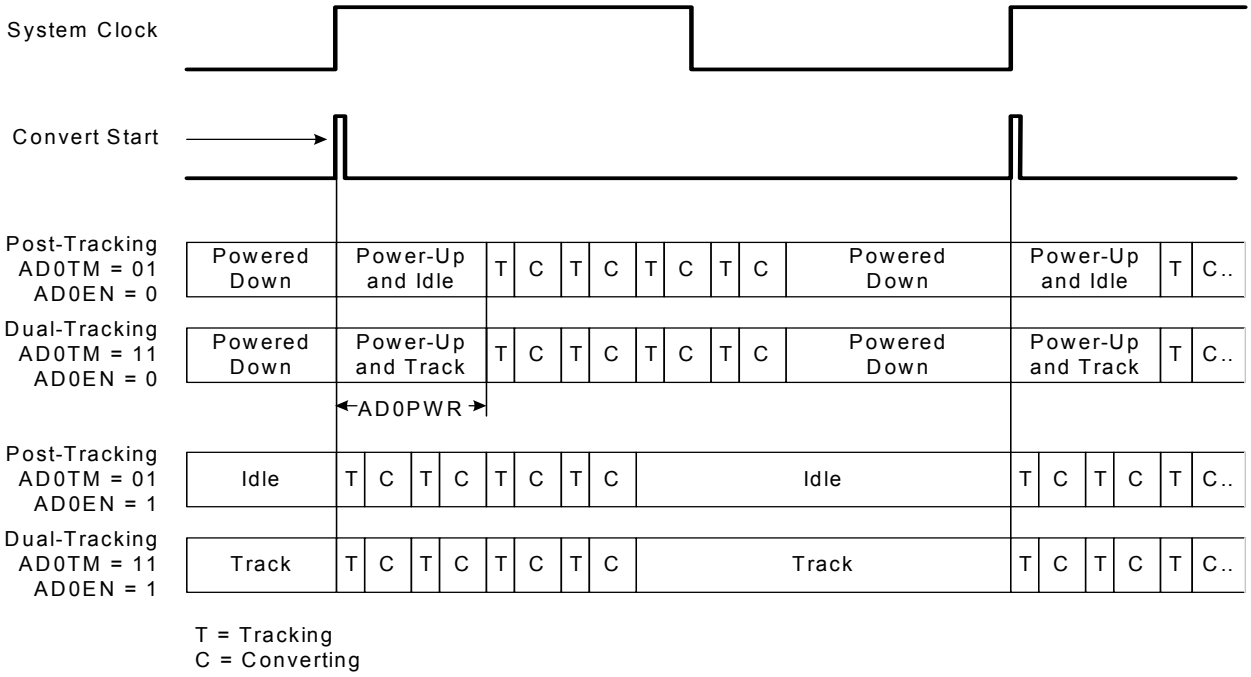


Figure 5.5. 12-Bit ADC Burst Mode Example with Repeat Count Set to 4

5.3.5. Output Conversion Code

The registers ADC0H and ADC0L contain the high and low bytes of the output conversion code. When the repeat count is set to 1, conversion codes are represented in 12-bit unsigned integer format and the output conversion code is updated after each conversion. Inputs are measured from '0' to V_{REF} x 4095/4096. Data can be right-justified or left-justified, depending on the setting of the AD0LJST bit (ADC0CN.2). Unused bits in the ADC0H and ADC0L registers are set to '0'. Example codes are shown below for both right-justified and left-justified data.

Input Voltage	Right-Justified ADC0H:ADC0L (AD0LJST = 0)	Left-Justified ADC0H:ADC0L (AD0LJST = 1)
V_{REF} x 4095/4096	0x0FFF	0xFFF0
V_{REF} x 2048/4096	0x0800	0x8000
V_{REF} x 2047/4096	0x07FF	0x7FF0
0	0x0000	0x0000

When the ADC0 Repeat Count is greater than 1, the output conversion code represents the accumulated result of the conversions performed and is updated after the last conversion in the series is finished. The output value can be 14-bit (4 samples), 15-bit (8 samples), or 16-bit (16 samples) in unsigned integer format based on the selected repeat count. The repeat count can be selected using the AD0RPT bits in the ADC0CF register. The value must be right-justified (AD0LJST = "0"), and unused bits in the ADC0H and ADC0L registers are set to '0'. The following example shows right-justified codes for repeat counts greater than 1. Notice that accumulating 2^n samples is equivalent to left-shifting by *n* bit positions when all samples returned from the ADC have the same value.

Input Voltage	Repeat Count = 4	Repeat Count = 8	Repeat Count = 16
V_{REF} x 4095/4096	0x3FFC	0x7FF8	0xFFF0
V_{REF} x 2048/4096	0x2000	0x4000	0x8000
V_{REF} x 2047/4096	0x1FFC	0x3FF8	0x7FF0
0	0x0000	0x0000	0x0000

5.3.6. Settling Time Requirements

A minimum tracking time is required before an accurate conversion can be performed. This tracking time is determined by the AMUX0 resistance, the ADC0 sampling capacitance, any external source resistance, and the accuracy required for the conversion.

Figure 5.6 shows the equivalent ADC0 input circuit. The required ADC0 settling time for a given settling accuracy (SA) may be approximated by Equation 5.1. When measuring the Temperature Sensor output or V_{DD} with respect to GND, R_{TOTAL} reduces to R_{MUX}. See Table 5.1 and Table 5.2 for ADC0 minimum settling time requirements.

$$t = \ln\left(\frac{2^n}{SA}\right) \times R_{TOTAL} C_{SAMPLE}$$

Equation 5.1. ADC0 Settling Time Requirements

Where:
SA is the settling accuracy, given as a fraction of an LSB (for example, 0.25 to settle within 1/4 LSB)
t is the required settling time in seconds
R_{TOTAL} is the sum of the AMUX0 resistance and any external source resistance.
n is the ADC resolution in bits (12).

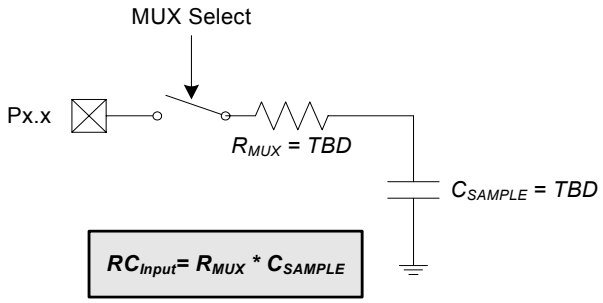


Figure 5.6. ADC0 Equivalent Input Circuits

SFR Definition 5.1. ADC0MX: ADC0 Channel Select

R	R	R	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	AD0MX					00011111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xBB

Bits7–5: **UNUSED**. Read = 000b; Write = don't care.
Bits4–0: **AD0MX4–0**: AMUX0 Positive Input Selection

AD0MX4–0	ADC0 Input Channel
00000	P0.0
00001	P0.1
00010	P0.2
00011	P0.3
00100	P0.4
00101	P0.5
00110	P0.6*
00111	P0.7*
01000	P1.0*
01001	P1.1*
01010	P1.2*
01011	P1.3*
01100	P1.4*
01101	P1.5*
01110	P1.6*
01111	P1.7*
11000	Temp Sensor
11001	V_{DD}
11010 - 11111	GND

***Note:** Only applies to C8051F53x parts.

SFR Definition 5.2. ADC0CF: ADC0 Configuration

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
AD0SC					AD0RPT		ATTEN	11111000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xBC

Bits7–3: **AD0SC4–0**: ADC0 SAR Conversion Clock Period Bits.
SAR Conversion clock is derived from FCLK by the following equation, where *AD0SC* refers to the 5-bit value held in bits AD0SC4**–**0. SAR Conversion clock requirements are given in Table 5.1.
BURSTEN = 0: FCLK is the current system clock.
BURSTEN = 1: FCLK is a maximum of 25 MHz, independent of the current system clock.

$$AD0SC = \frac{FCLK}{CLK_{SAR}} - 1^{*} \quad \text{or} \quad CLK_{SAR} = \frac{FCLK}{AD0SC + 1}$$

***Note:** Round the result up.

Bits2–1: **AD0RPT1–0**: ADC0 Repeat Count.
Controls the number of conversions taken and accumulated between ADC0 End of Conversion (ADCINT) and ADC0 Window Comparator (ADCWINT) interrupts. A convert start is required for each conversion unless Burst Mode is enabled. In Burst Mode, a single convert start can initiate multiple self-timed conversions. Results in both modes are accumulated in the ADC0H:ADC0L register. **When AD0RPT1–0 are set to a value other than '00', the AD0LJST bit in the ADC0CN register must be set to '0' (right justified).**
00: 1 conversion is performed.
01: 4 conversions are performed and accumulated.
10: 8 conversions are performed and accumulated.
11: 16 conversions are performed and accumulated.

Bit0: **ATTEN**: Attenuation Enabled Bit.
Controls the attenuation programming. For more information of the usage please refer to the following chapter: Section "**5.5. Selectable Attenuation**" on page **57**.

SFR Definition 5.3. ADC0H: ADC0 Data Word MSB

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xBE

Bits7–0: ADC0 Data Word High-Order Bits.
For AD0LJST = 0 and AD0RPT as follows:
00: Bits 3–0 are the upper 4 bits of the 12-bit result. Bits 7–4 are 0000b.
01: Bits 4–0 are the upper 5 bits of the 14-bit result. Bits 7–5 are 000b.
10: Bits 5–0 are the upper 6 bits of the 15-bit result. Bits 7–6 are 00b.
11: Bits 7–0 are the upper 8 bits of the 16-bit result.
For AD0LJST = 1 (AD0RPT must be '00'): Bits 7–0 are the most-significant bits of the ADC0 12-bit result.

SFR Definition 5.4. ADC0L: ADC0 Data Word LSB

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xBD

Bits7–0: ADC0 Data Word Low-Order Bits.
For AD0LJST = 0: Bits 7–0 are the lower 8 bits of the ADC0 Accumulated Result.
For AD0LJST = 1 (AD0RPT must be '00'): Bits 7–4 are the lower 4 bits of the 12-bit result.
Bits 3–0 are 0000b.

SFR Definition 5.5. ADC0CN: ADC0 Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
AD0EN	BURSTEN	AD0INT	AD0BUSY	AD0WINT	AD0LJST	AD0CM1	AD0CM0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xE8

Bit7: **AD0EN**: ADC0 Enable Bit.
0: ADC0 Disabled. ADC0 is in low-power shutdown.
1: ADC0 Enabled. ADC0 is active and ready for data conversions.

Bit6: **BURSTEN**: ADC0 Burst Mode Enable Bit.
0: ADC0 Burst Mode Disabled.
1: ADC0 Burst Mode Enabled.

Bit5: **AD0INT**: ADC0 Conversion Complete Interrupt Flag.
0: ADC0 has not completed a data conversion since the last time AD0INT was cleared.
1: ADC0 has completed a data conversion.

Bit4: **AD0BUSY**: ADC0 Busy Bit.
Read:
0: ADC0 conversion is complete or a conversion is not currently in progress. AD0INT is set to logic 1 on the falling edge of AD0BUSY.
1: ADC0 conversion is in progress.
Write:
0: No Effect.
1: Initiates ADC0 Conversion if AD0CM2–0 = 000b

Bit3: **AD0WINT**: ADC0 Window Compare Interrupt Flag.
This bit must be cleared by software.
0: ADC0 Window Comparison Data match has not occurred since this flag was last cleared.
1: ADC0 Window Comparison Data match has occurred.

Bit2: **AD0LJST**: ADC0 Left Justify Select
0: Data in ADC0H:ADC0L registers is right justified.
1: Data in ADC0H:ADC0L registers is left justified. This option should not be used with a repeat count greater than 1 (when AD0RPT1–0 is 01b, 10b, or 11b).

Bits1–0: **AD0CM1–0**: ADC0 Start of Conversion Mode Select.
00: ADC0 conversion initiated on every write of ‘1’ to AD0BUSY.
01: ADC0 conversion initiated on overflow of Timer 3.
10: ADC0 conversion initiated on rising edge of external CNVSTR.
11: ADC0 conversion initiated on overflow of Timer 2.

SFR Definition 5.6. ADC0TK: ADC0 Tracking Mode Select

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
AD0PWR				AD0TM		AD0TK		11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
							(bit addressable)	0xBA

Bits7–4: **AD0PWR3–0**: ADC0 Burst Power-Up Time.
For BURSTEN = 0:
ADC0 power state controlled by AD0EN.
For BURSTEN = 1 and AD0EN = 1;
ADC0 remains enabled and does not enter the very low power state.
For BURSTEN = 1 and AD0EN = 0:
ADC0 enters the very low power state as specified in Table 5.1 and Table 5.2 and is enabled after each convert start signal. The Power Up time is programmed according to the following equation:

$$AD0PWR = \frac{Tstartup}{200ns} - 1 \quad \text{or} \quad Tstartup = (AD0PWR + 1)200ns$$

Bits3–2: **AD0TM1–0**: ADC0 Tracking Mode Select Bits.
00: Reserved.
01: ADC0 is configured to Post-Tracking Mode.
10: ADC0 is configured to Pre-Tracking Mode.
11: ADC0 is configured to Dual-Tracking Mode (default).

Bits1–0: **AD0TK1–0**: ADC0 Post-Track Time.
Post-Tracking time is controlled by AD0TK as follows:
00: Post-Tracking time is equal to 2 SAR clock cycles + 2 FCLK cycles.
01: Post-Tracking time is equal to 4 SAR clock cycles + 2 FCLK cycles.
10: Post-Tracking time is equal to 8 SAR clock cycles + 2 FCLK cycles.
11: Post-Tracking time is equal to 16 SAR clock cycles + 2 FCLK cycles.

5.4. Programmable Window Detector

The ADC Programmable Window Detector continuously compares the ADC0 output registers to user-programmed limits, and notifies the system when a desired condition is detected. This is especially effective in an interrupt-driven system, saving code space and CPU bandwidth while delivering faster system response times. The window detector interrupt flag (AD0WINT in register ADC0CN) can also be used in polled mode. The ADC0 Greater-Than (ADC0GTH, ADC0GTL) and Less-Than (ADC0LTH, ADC0LTL) registers hold the comparison values. The window detector flag can be programmed to indicate when measured data is inside or outside of the user-programmed limits, depending on the contents of the ADC0 Less-Than and ADC0 Greater-Than registers.

SFR Definition 5.7. ADC0GTH: ADC0 Greater-Than Data High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xC4

Bits7–0: High byte of ADC0 Greater-Than Data Word.

SFR Definition 5.8. ADC0GTL: ADC0 Greater-Than Data Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xC3

Bits7–0: Low byte of ADC0 Greater-Than Data Word.

SFR Definition 5.9. ADC0LTH: ADC0 Less-Than Data High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xC6

Bits7–0: High byte of ADC0 Less-Than Data Word.

SFR Definition 5.10. ADC0LTL: ADC0 Less-Than Data Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xC5

Bits7–0: Low byte of ADC0 Less-Than Data Word.

5.4.1. Window Detector In Single-Ended Mode

Figure 5.7 shows two example window comparisons for right-justified data with ADC0LTH:ADC0LTL = 0x0200 (512d) and ADC0GTH:ADC0GTL = 0x0100 (256d). The input voltage can range from '0' to V_{REF} x (4095/4096) with respect to GND, and is represented by a 12-bit unsigned integer value. The repeat count is set to one. In the left example, an AD0WINT interrupt will be generated if the ADC0 conversion word (ADC0H:ADC0L) is within the range defined by ADC0GTH:ADC0GTL and ADC0LTH:ADC0LTL (if 0x0100 < ADC0H:ADC0L < 0x0200). In the right example, and AD0WINT interrupt will be generated if the ADC0 conversion word is outside of the range defined by the ADC0GT and ADC0LT registers (if ADC0H:ADC0L < 0x0100 or ADC0H:ADC0L > 0x0200). Figure 5.8 shows an example using left-justified data with the same comparison values.

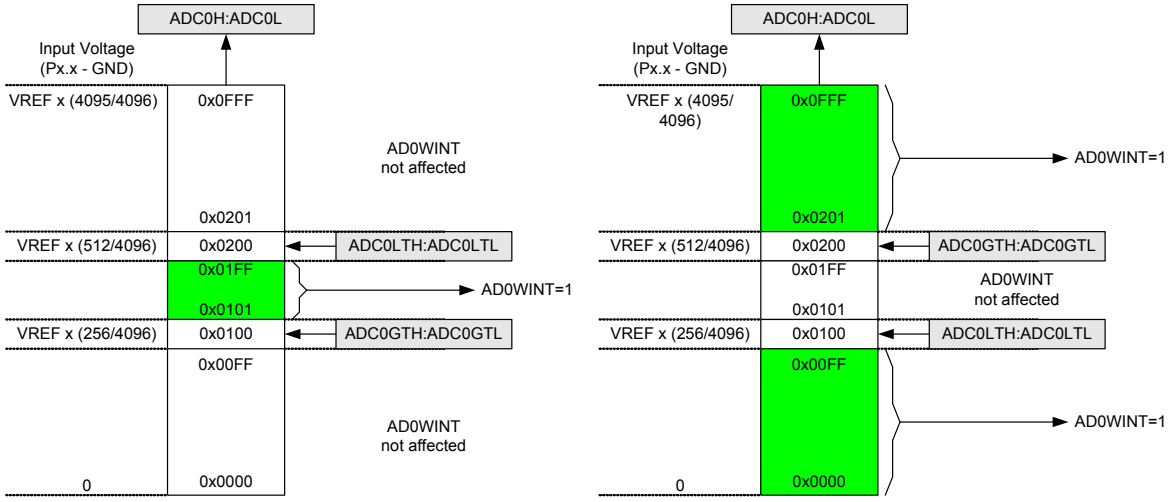


Figure 5.7. ADC Window Compare Example: Right-Justified Single-Ended Data

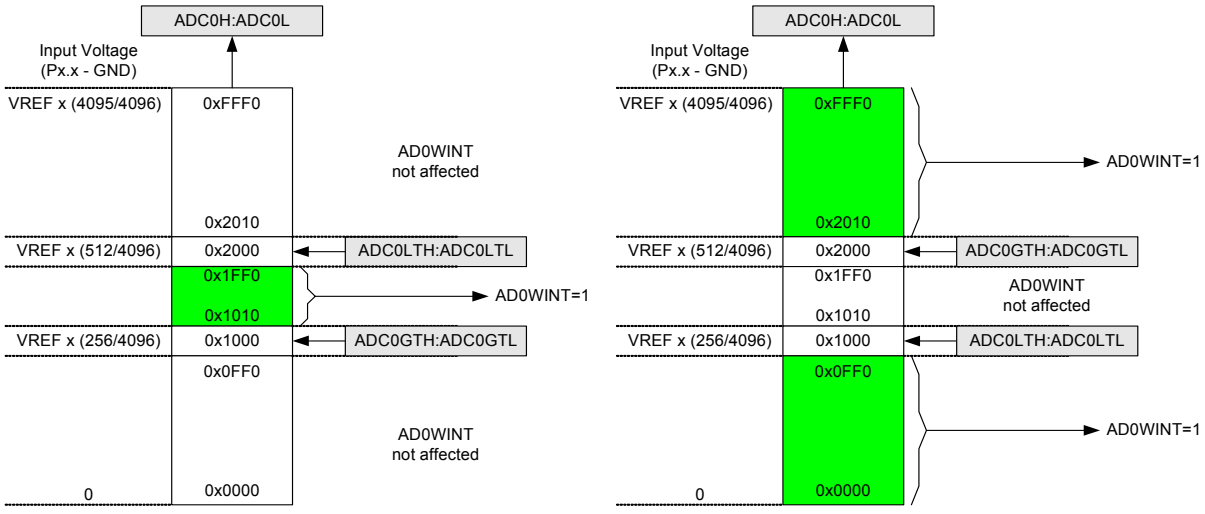


Figure 5.8. ADC Window Compare Example: Left-Justified Single-Ended Data

5.5. Selectable Attenuation

The C8051F52x/F53x family of devices implements an ADC that provides a new and innovative selectable attenuation option. This option allows the designer to take the ADC Input and either keep its input value unchanged or attenuate by a factor of 2 (value divided by two).

The attenuation selection is performed using the following steps:

Step 1. Set the ATTEN bit (ADC0CF.0)
Step 2. Load the ADC0H with 0x04
Step 3. Load ADC0L with 0xFC if no attenuation (1/1 gain) is required or 0x7C to attenuate the signal (1/2 gain)
Step 4. Reset the ATTEN bit (ADC0CF.0)

Notes:

1. During the Attenuation selection no ADC conversion should be performed as the results will be incorrect.
2. The maximum input voltage value is still limited to Vregin and the maximum value of the signal after attenuation is limited to Vref otherwise the ADC will saturate.

Table 5.1. ADC0 Electrical Characteristics (V_{DD} = 2.4 V, V_{REF} = 2.25 V)

V_{DD} = 2.4 V, V_{REF} = 2.25 V (REFSL=0), –40 to +125 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
DC Accuracy					
Resolution			12		bits
Integral Nonlinearity	C8051F52x/C8051F53x devices 	— —	— —	±1 ±2	LSB
Differential Nonlinearity	Guaranteed Monotonic	—	—	±1	LSB
Offset Error		—	TBD	—	LSB
Full Scale Error	Differential mode	—	TBD	—	LSB
Offset Temperature Coefficient		—	TBD	—	ppm/°C
Dynamic Performance (10 kHz sine-wave Single-ended input, 0 to 1 dB below Full Scale, 200 ksps)					
Signal-to-Noise Plus Distortion	C8051F52x/C8051F53x devices 	68 64	— —	— —	dB
Total Harmonic Distortion	Up to the 5th harmonic	—	TBD	—	dB
Spurious-Free Dynamic Range		—	TBD	—	dB
Conversion Rate					
SAR Conversion Clock		—	—	10	MHz
Conversion Time in SAR Clocks[1]		—	13	—	clocks
Track/Hold Acquisition Time[2]		1	—	—	µs
Throughput Rate		—	—	200	ksps
Analog Inputs					
Input Voltage Range[3]		0	—	4.6 or 2.3	V
Input Capacitance		—	TBD	—	pF
Temperature Sensor					
Linearity[4,5]		—	±TBD	—	°C
Gain[4,5]		—	TBD	—	µV / °C
Offset[4,5]	(Temp = 0 °C)	—	TBD	—	mV
Power Specifications					
Power Supply Current (V_{DD} supplied to ADC)	Operating Mode, 200 ksps	—	680	TBD	µA
Burst Mode (Idle)		—	TBD	—	µA
Power-On Time		TBD	—	—	µs
Power Supply Rejection		—	TBD	—	mV/V

Notes:

1. An additional 2 FCLK cycles are required to start and complete a conversion.
2. Additional tracking time may be required depending on the output impedance connected to the ADC input. See Section "**5.3.6. Settling Time Requirements**" on page **48**.
3. The maximum input voltage is 2.3 V without attenuation and 4.6 V with attenuation when using the internal reference. If an external reference is used then the input is limited to the external reference value.
4. Represents one standard deviation from the mean.
5. Includes ADC offset, gain, and linearity variations.

Table 5.2. ADC0 Electrical Characteristics (V_{DD} = 2.0 V, V_{REF} = 1.5 V)

V_{DD} = 2.0 V, V_{REF} = 1.5 V (REFSL = 0), –40 to +125 °C unless otherwise specified

Parameter	Conditions	Min	Typ	Max	Units
DC Accuracy					
Resolution			12		bits
Integral Nonlinearity	C8051F52x/C8051F53x devices	— —	— —	±1 ±2	LSB
Differential Nonlinearity	Guaranteed Monotonic	—	—	±1	LSB
Offset Error		—	TBD	—	LSB
Full Scale Error	Differential mode	—	TBD	—	LSB
Offset Temperature Coefficient		—	TBD	—	ppm/°C
Dynamic Performance (10 kHz sine-wave Single-ended input, 0 to 1 dB below Full Scale, 200 ksps)					
Signal-to-Noise Plus Distortion	C8051F52x/C8051F53x devices	68 64	— —	— —	dB
Total Harmonic Distortion	Up to the 5th harmonic	—	TBD	—	dB
Spurious-Free Dynamic Range		—	TBD	—	dB
Conversion Rate					
SAR Conversion Clock		—	—	10	MHz
Conversion Time in SAR Clocks	Note 1	—	13	—	clocks
Track/Hold Acquisition Time	Note 2	1	—	—	µs
Throughput Rate		—	—	200	ksps
Analog Inputs					
Input Voltage Range		0	—	V_{REF}	V
Input Capacitance		—	TBD	—	pF
Temperature Sensor					
Linearity	Notes 3, 4	—	±TBD	—	°C
Gain	Notes 3, 4	—	TBD	—	µV / °C
Offset	Notes 3, 4 (Temp = 0 °C)	—	TBD	—	mV
Power Specifications					
Power Supply Current (V_{DD} supplied to ADC0)	Operating Mode, 200 ksps	—	650	TBD	µA
Burst Mode (Idle)		—	TBD	—	µA
Power-On Time		TBD	—	—	µs
Power Supply Rejection		—	TBD	—	mV/V

Notes:
1. An additional 2 FCLK cycles are required to start and complete a conversion.
2. Additional tracking time may be required depending on the output impedance connected to the ADC input. See Section "**5.3.6. Settling Time Requirements**" on page **48**.
3. Represents one standard deviation from the mean.
4. Includes ADC offset, gain, and linearity variations.

NOTES:

6. Voltage Reference

The Voltage reference MUX on C8051F52x/F53x devices is configurable to use an externally connected voltage reference, the internal reference voltage generator, or the V_{DD} power supply voltage (see Figure 6.1). The REFSL bit in the Reference Control register (REF0CN) selects the reference source. For an external source or the internal reference applied to the V_{REF} pin, REFSL should be set to '0'. To use V_{DD} as the reference source, REFSL should be set to '1'.

The BIASE bit enables the internal voltage bias generator, which is used by the ADC, Temperature Sensor, and internal oscillators. This bit is forced to logic 1 when any of the aforementioned peripherals are enabled. The bias generator may be enabled manually by writing a '1' to the BIASE bit in register REF0CN; see SFR Definition 6.1 for REF0CN register details. The electrical specifications for the voltage reference circuit are given in Table 6.1.

The internal voltage reference circuit consists of a temperature stable bandgap voltage reference generator and a gain-of-two output buffer amplifier. The output voltage is selectable between 1.5 V and 2.25 V. The internal voltage reference can be driven out on the V_{REF} pin by setting the REFBE bit in register REF0CN to a '1' (see Figure 6.1). The load seen by the V_{REF} pin must draw less than 200 µA to GND. When using the internal voltage reference, bypass capacitors of 0.1 µF and 4.7 µF are recommended from the V_{REF} pin to GND. If the internal reference is not used, the REFBE bit should be cleared to '0'. Electrical specifications for the internal voltage reference are given in Table 6.1.

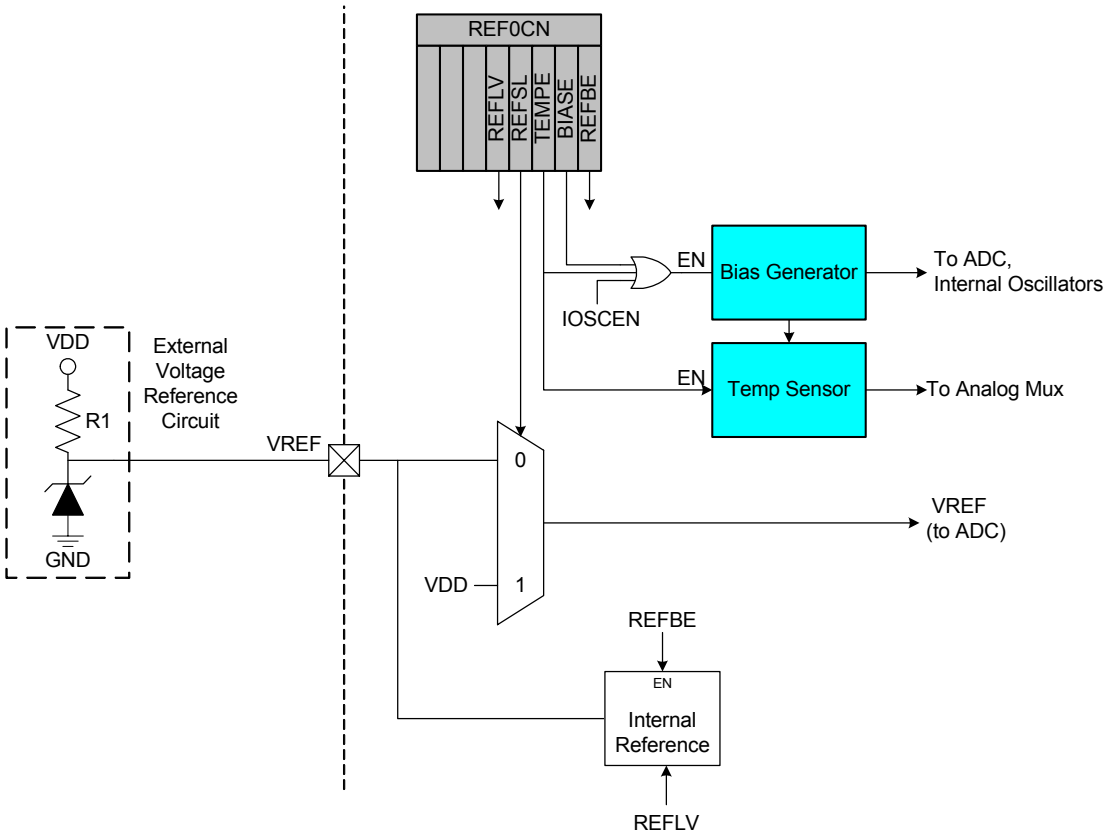


Figure 6.1. Voltage Reference Functional Block Diagram

Important Note About the V_{REF} Pin: Port pin P0.0 is used as the external V_{REF} input and as an output for the internal V_{REF}. When using either an external voltage reference or the internal reference circuitry, P0.0 should be configured as an analog pin, and skipped by the Digital Crossbar. To configure P0.0 as an analog pin, clear Bit 2 in register P0MDIN to '0'. To configure the Crossbar to skip P0.0, set Bit 0 in register P0SKIP to '1'. Refer to **Section "14. Port Input/Output" on page 115** for complete Port I/O configuration details. The TEMPE bit in register REF0CN enables/disables the temperature sensor. While disabled, the temperature sensor defaults to a high impedance state and any ADC0 measurements performed on the sensor result in meaningless data.

SFR Definition 6.1. REF0CN: Reference Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
—	—	ZTCEN	REFLV	REFSL	TEMPE	BIASE	REFBE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0xD1

Bits7–6: **RESERVED**. Read = 0b. Must write 0b.

Bit5: **ZTCEN**: Zero-TempCo Bias Enable Bit.
0: ZeroTC Bias Generator automatically enabled when needed.
1: ZeroTC Bias Generator forced on.

Bit4: **REFLV**: Voltage Reference Output Level Select.
This bit selects the output voltage level for the internal voltage reference.
0: Internal voltage reference set to 1.5 V.
1: Internal voltage reference set to 2.25 V.

Bit3: **REFSL**: Voltage Reference Select.
This bit selects the source for the internal voltage reference.
0: V_{REF} pin used as voltage reference.
1: V_{DD} used as voltage reference.

Bit2: **TEMPE**: Temperature Sensor Enable Bit.
0: Internal Temperature Sensor off.
1: Internal Temperature Sensor on.

Bit1: **BIASE**: Internal Analog Bias Generator Enable Bit.
0: Internal Analog Bias Generator automatically enabled when needed.
1: Internal Analog Bias Generator on.

Bit0: **REFBE**: Internal Reference Buffer Enable Bit.
0: Internal Reference Buffer disabled.
1: Internal Reference Buffer enabled. Internal voltage reference driven on the V_{REF} pin.

Table 6.1. Voltage Reference Electrical Characteristics

V_{DD} = 2.0 V; –40 to +125 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Internal Reference (REFBE = 1)					
Output Voltage	25 °C ambient (REFLV = 0) 25 °C ambient (REFLV = 1), V_{DD} = 2.4 V	TBD TBD	1.5 2.2	TBD TBD	V
V_{REF} Short-Circuit Current		—	—	TBD	mA
V_{REF} Temperature Coefficient		—	TBD	—	ppm/°C
Power Consumption (Internal)		—	TBD	—	µA
Load Regulation	Load = 0 to 200 µA to GND	—	TBD	—	ppm/µA
V_{REF} Turn-on Time 1	4.7 µF tantalum, 0.1 µF ceramic bypass	—	TBD	—	ms
V_{REF} Turn-on Time 2	no bypass cap	—	TBD	—	µs
Power Supply Rejection		—	TBD	—	ppm/V
External Reference (REFBE = 0)					
Input Voltage Range		0	—	V_{DD}	V
Input Current	Sample Rate = 200 ksps; V_{REF} = TBD V	—	TBD	—	µA
Bias Generators					
ADC Bias Generator	BIASE = ‘1’	—	TBD	—	µA
Reference Bias Generator		—	TBD	—	µA

NOTES:

7. Voltage Regulator (REG0)

C8051F52x/F53x devices include an on-chip low dropout voltage regulator (REG0). The input to REG0 at the V_{REGIN} pin can be as high as 5.25 V. The output can be selected by software to 2.0 V or 2.4 V. When enabled, the output of REG0 appears on the V_{DD} pin, powers the microcontroller core, and can be used to power external devices. On reset, REG0 is enabled and can be disabled by software.

The input (V_{REGIN}) and output (V_{DD}) of the voltage regulator should both be bypassed with a large capacitor (4.7 µF + 0.1 µF) to ground. This capacitor will eliminate power spikes and provide any immediate power required by the microcontroller. The settling time associated with the voltage regulator is shown in Table 7.1.

The Voltage regulator can also generate an interrupt (if enabled by EREG0, EIE1.6) that is triggered whenever the Vregin Input voltage drops below the dropout threshold. (see Table 7.1)

This dropout interrupt has no pending flag and the recommended procedure to use it is as follows:

Step 1. Wait enough time to ensure the Vregin input voltage is stable
Step 2. Enable the dropout interrupt (EREG0, EIE1.6) and select the proper priority (PREG0, PIE1.6)
Step 3. If triggered, inside the interrupt disable it (clear EREG0, EIE1.6), execute all procedures necessary to protect your application (put it in a safe mode and leave the interrupt now disabled.
Step 4. In the main application, now running in the safe mode, regularly checks the DROPOUT bit (REG0CN.0). Once it is cleared by the regulator hardware the application can enable the interrupt again (EREG0, EIE1.6) and return to the normal mode operation.

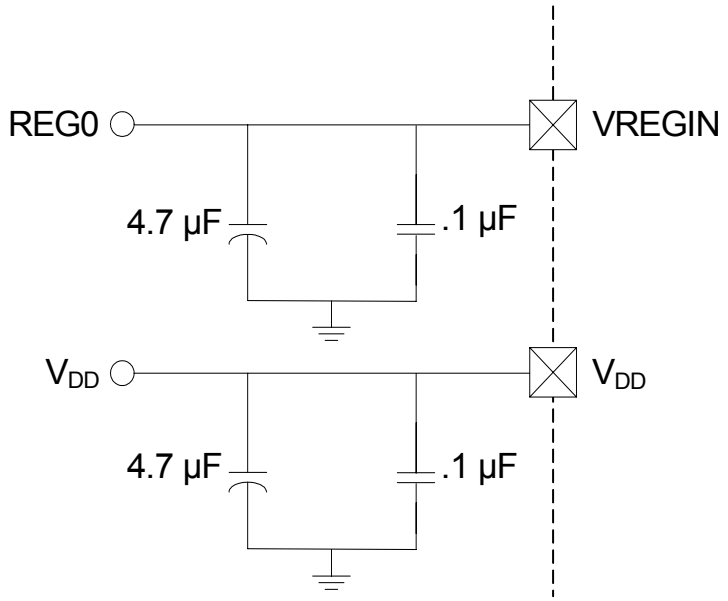


Figure 7.1. External Capacitors for Voltage Regulator Input/Output

SFR Definition 7.1. REG0CN: Regulator Control

R/W	R/W	R	R/W	R	R	R	R	Reset Value
REGDIS	Reserved	—	REG0MD	—	—	—	DROPOUT	00010000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xC9

Bit7: **REGDIS**: Voltage Regulator Disable Bit.
This bit disables/enables the Voltage Regulator.
0: Voltage Regulator Enabled.
1: Voltage Regulator Disabled.
Bit6: **RESERVED**. Read = 0b. Must write 0b.
Bit5: **UNUSED**. Read = 0b. Write = don't care.
Bit4: **REG0MD**: Voltage Regulator Mode Select Bit.
This bit selects the Voltage Regulator output voltage.
0: Voltage Regulator output is 2.0 V.
1: Voltage Regulator output is 2.4 V (default).
Bits3–1: **UNUSED**. Read = 0b. Write = don't care.
Bit0: **DROPOUT**: Voltage Regulator Dropout Indicator Bit.
0: Voltage Regulator is not in dropout.
1: Voltage Regulator is in or near dropout.

Table 7.1. Voltage Regulator Electrical Specifications

V_{DD} = 2.0 or 2.4 V; –40 to +125 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
Input Voltage Range (V_{REGIN})*		2.7*	—	5.25	V
Dropout Voltage (V_{DO})	Output Current = 1 mA Output Current = 50 mA	TBD TBD	10 500	TBD TBD	mV
Output Voltage (V_{DD})	2.0 V operation (REG0MD = '0') 2.0 V operation (REG0MD = '1') Output Current = 1 to 50 mA	TBD TBD —	2.0 2.4 —	TBD TBD —	V
Bias Current	2.0 V operation (REG0MD = '0') 2.4 V operation (REG0MD = '1')	— —	1 1	TBD TBD	µA
Dropout Indicator Detection Threshold		TBD	—	TBD	V
Output Voltage Tempco		—	TBD	—	mV/ºC
VREG Settling Time	50 mA load with V_{REGIN} = 2.4 V and V_{DD} load capacitor of 4.8 µF	—	250	—	µs

***Note:** The minimum input voltage is 2.7 V or V_{DD} + V_{DO}(max load), whichever is greater.

8. Comparator

C8051F52x/F53x devices include one on-chip programmable voltage comparator. The Comparator is shown in Figure 8.1;

The Comparator offers programmable response time and hysteresis, an analog input multiplexer, and two outputs that are optionally available at the Port pins: a synchronous “latched” output (CP0), or an asynchronous “raw” output (CP0A). The asynchronous CP0A signal is available even when the system clock is not active. This allows the Comparator to operate and generate an output with the device in STOP or SUSPEND mode. When assigned to a Port pin, the Comparator output may be configured as open drain or push-pull (see **Section “14.2. Port I/O Initialization” on page 121**). The Comparator may also be used as a reset source (see **Section “12.5. Comparator Reset” on page 100**).

The Comparator inputs are selected in the CPT0MX register (SFR Definition 8.2). The CMX0P3–CMX0P0 bits select the Comparator0 positive input; the CMX0N3–CMX0N0 bits select the Comparator0 negative input.

Important Note About Comparator Inputs: The Port pins selected as Comparator inputs should be configured as analog inputs in their associated Port configuration register, and configured to be skipped by the Crossbar (for details on Port configuration, see **Section “14.3. General Purpose Port I/O” on page 123**)

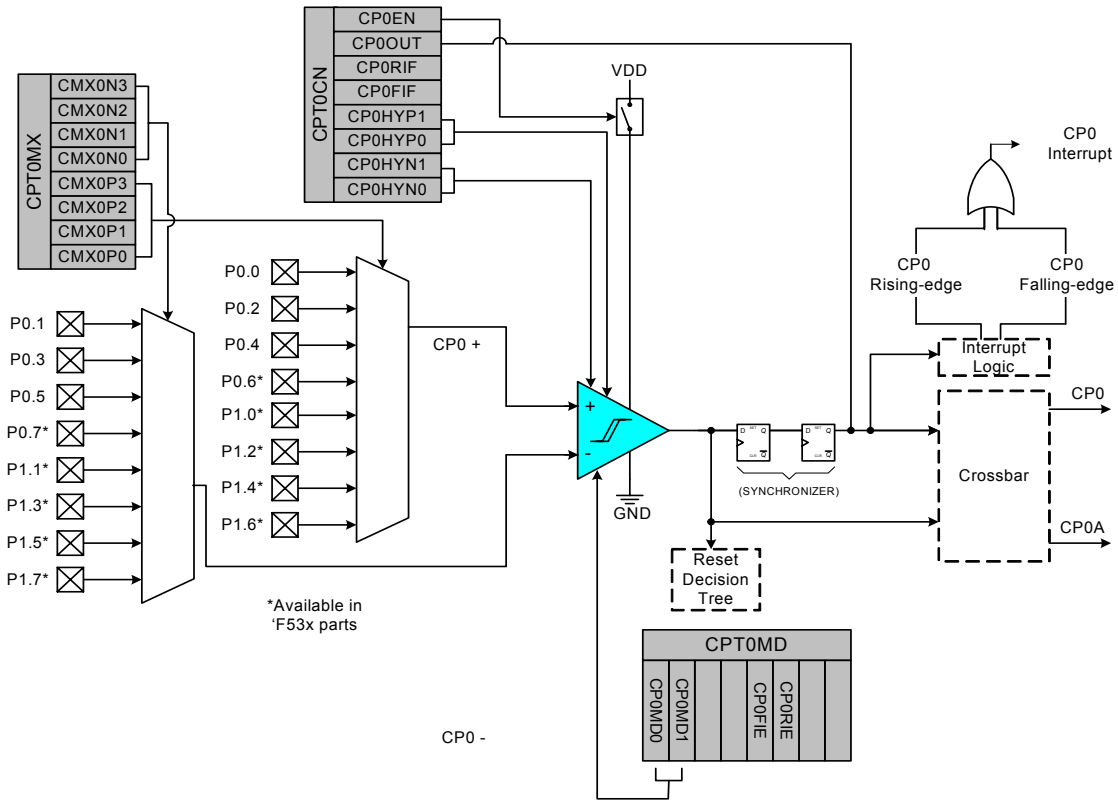


Figure 8.1. Comparator Functional Block Diagram

The Comparator has two input modes: Low-Speed Analog Mode and High-Speed Analog Mode. The difference between the two modes is that Comparator input resistance is decreased in High-Speed Analog

Mode, but power consumption is slightly increased. High-Speed Analog Mode is enabled by setting the CPnHIQE bit in CPTnMD.

The Comparator output can be polled in software, used as an interrupt source, internal oscillator suspend awakening source and/or routed to a Port pin. When routed to a Port pin, the Comparator output is available asynchronous or synchronous to the system clock; the asynchronous output is available even in STOP or SUSPEND mode (with no system clock active). When disabled, the Comparator output (if assigned to a Port I/O pin via the Crossbar) defaults to the logic low state, and its supply current falls to less than 100 nA. See **Section "14.1. Priority Crossbar Decoder" on page 117** for details on configuring Comparator outputs via the digital Crossbar. Comparator inputs can be externally driven from –0.25 V to (V_{DD}) + 0.25 V without damage or upset. The complete Comparator electrical specifications are given in Table 8.1.

The Comparator response time may be configured in software via the CPTnMD register (see SFR Definition 8.3). Selecting a longer response time reduces the Comparator supply current. See Table 8.1 for complete timing and current consumption specifications.

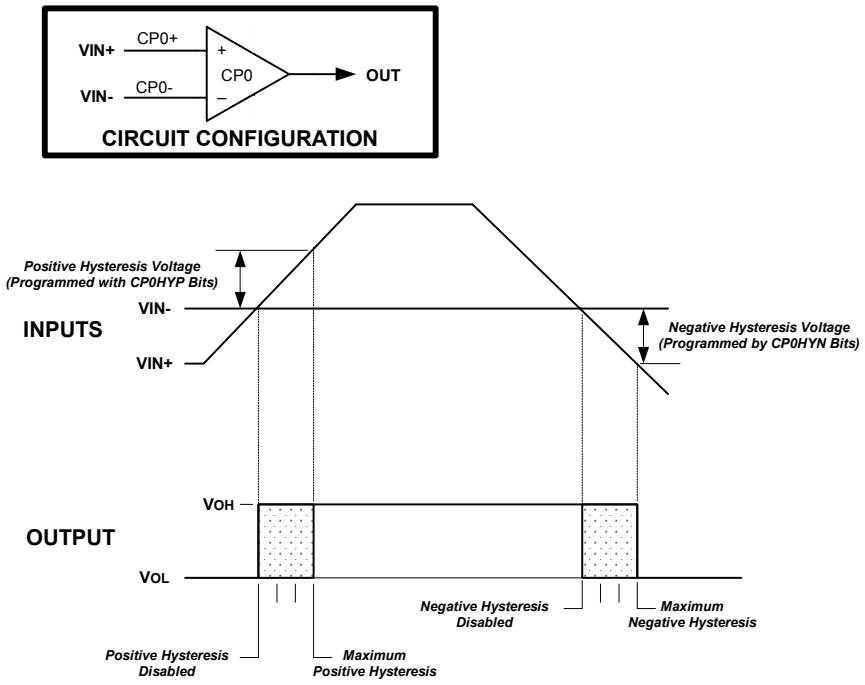


Figure 8.2. Comparator Hysteresis Plot

The Comparator hysteresis is software-programmable via its Comparator Control register CPT0CN (for n = 0 or 1). The user can program both the amount of hysteresis voltage (referred to the input voltage) and the positive and negative-going symmetry of this hysteresis around the threshold voltage.

The Comparator hysteresis is programmed using Bits3–0 in the Comparator Control Register CPT0CN (shown in SFR Definition 8.1). The amount of negative hysteresis voltage is determined by the settings of the CP0HYN bits. As shown in Table 8.1, settings of 20, 10 or 5 mV of negative hysteresis can be

programmed, or negative hysteresis can be disabled. In a similar way, the amount of positive hysteresis is determined by the setting the CP0HYP bits.

Comparator interrupts can be generated on both rising-edge and falling-edge output transitions. (For Interrupt enable and priority control, see **Section "11. Interrupt Handler" on page 89**). The CP0FIF flag is set to logic 1 upon a Comparator falling-edge detect, and the CPnRIF flag is set to logic 1 upon the Comparator rising-edge detect. Once set, these bits remain set until cleared by software. The output state of the Comparator can be obtained at any time by reading the CPnOUT bit. The Comparator is enabled by setting the CP0EN bit to logic 1, and is disabled by clearing this bit to logic 0.

The output state of the Comparator can be obtained at any time by reading the CP0OUT bit. The Comparator is enabled by setting the CP0EN bit to logic 1, and is disabled by clearing this bit to logic 0. When the Comparator is enabled, the internal oscillator is awakened from SUSPEND mode if the Comparator output is logic 0.

Note that false rising edges and falling edges can be detected when the comparator is first powered-on or if changes are made to the hysteresis or response time control bits. Therefore, it is recommended that the rising-edge and falling-edge flags be explicitly cleared to logic 0 a short time after the comparator is enabled or its mode bits have been changed. This Power Up Time is specified in Table 8.1 on page 72.

SFR Definition 8.1. CPT0CN: Comparator0 Control

R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CP0EN	CP0OUT	CP0RIF	CP0FIF	CP0HYP1	CP0HYP0	CP0HYN1	CP0HYN0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0x9B

Bit7: **CP0EN**: Comparator0 Enable Bit.
0: Comparator0 Disabled.
1: Comparator0 Enabled.

Bit6: **CP0OUT**: Comparator0 Output State Flag.
0: Voltage on CP0+ < CP0–.
1: Voltage on CP0+ > CP0–.

Bit5: **CP0RIF**: Comparator0 Rising-Edge Flag.
0: No Comparator0 Rising Edge has occurred since this flag was last cleared.
1: Comparator0 Rising Edge has occurred.

Bit4: **CP0FIF**: Comparator0 Falling-Edge Flag.
0: No Comparator0 Falling-Edge has occurred since this flag was last cleared.
1: Comparator0 Falling-Edge has occurred.

Bits3–2: **CP0HYP1–0**: Comparator0 Positive Hysteresis Control Bits.
00: Positive Hysteresis Disabled.
01: Positive Hysteresis = 5 mV.
10: Positive Hysteresis = 10 mV.
11: Positive Hysteresis = 20 mV.

Bits1–0: **CP0HYN1–0**: Comparator0 Negative Hysteresis Control Bits.
00: Negative Hysteresis Disabled.
01: Negative Hysteresis = 5 mV.
10: Negative Hysteresis = 10 mV.
11: Negative Hysteresis = 20 mV.

SFR Definition 8.2. CPT0MX: Comparator0 MUX Selection

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CMX0N3	CMX0N2	CMX0N1	CMX0N0	CMX0P3	CMX0P2	CMX0P1	CMX0P0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x9F

Bits7–4: **CMX0N3–CMX0N0**: Comparator0 Negative Input MUX Select.
These bits select which Port pin is used as the Comparator0 negative input.

CMX0N3	CMX0N2	CMX0N1	CMX0N0	Negative Input
0	0	0	0	P0.1
0	0	0	1	P0.3
0	0	1	0	P0.5
0	0	1	1	P0.7*
0	1	0	0	P1.1*
0	1	0	1	P1.3*
0	1	1	0	P1.5*
0	1	1	1	P1.7*

***Note:** *Available only on the C8051F53x devices*

Bits1–0: **CMX0P3–CMX0P0**: Comparator0 Positive Input MUX Select.
These bits select which Port pin is used as the Comparator0 positive input.

CMX0P3	CMX0P2	CMX0P1	CMX0P0	Positive Input
0	0	0	0	P0.0
0	0	0	1	P0.2
0	0	1	0	P0.4
0	0	1	1	P0.6*
0	1	0	0	P1.0*
0	1	0	1	P1.2*
0	1	1	0	P1.4*
0	1	1	1	P1.6*

***Note:** *Available only on the C8051F53x devices.*

SFR Definition 8.3. CPT0MD: Comparator0 Mode Selection

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CP0HIQE	-	CP0RIE	CP0FIE	-	-	CP0MD1	CP0MD0	00000010
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x9D

Bit7: **CP0HIQE**: High-Speed Analog Mode Enable Bit.
0: Comparator input configured to Low-Speed Analog Mode.
1: Comparator input configured to High-Speed Analog Mode.
Bit6: **UNUSED**. Read = 0b. Write = don't care.
Bit5: **CP0RIE**: Comparator Rising-Edge Interrupt Enable.
0: Comparator rising-edge interrupt disabled.
1: Comparator rising-edge interrupt enabled.
Bit4: **CP0FIE**: Comparator Falling-Edge Interrupt Enable.
0: Comparator falling-edge interrupt disabled.
1: Comparator falling-edge interrupt enabled.
Note: It is necessary to enable both CP0xIE and the correspondent ECPx bit located in EIE1 SFR.
Bits3–2: **UNUSED**. Read = 00b. Write = don't care.
Bits1–0: **CP0MD1–CP0MD0**: Comparator0 Mode Select
These bits select the response time for Comparator0.

Mode	CP0MD1	CP0MD0	CP0 Falling Edge Response Time (TYP)
0	0	0	Fastest Response Time
1	0	1	—
2	1	0	—
3	1	1	Lowest Power Consumption

Note: Rising Edge response times are approximately double the Falling Edge response times.

Table 8.1. Comparator Electrical Characteristics

V_{DD} = 2.0 V, –40 to +125 °C unless otherwise noted.
All specifications apply to both Comparator0 and Comparator1 unless otherwise noted.

Parameter	Conditions	Min	Typ	Max	Units
Response Time: Mode 0, Vcm[1] = 1.5 V	CP0+ – CP0– = 100 mV	—	TBD	—	ns
	CP0+ – CP0– = –100 mV	—	TBD	—	ns
Response Time: Mode 1, Vcm[1] = 1.5 V	CP0+ – CP0– = 100 mV	—	TBD	—	ns
	CP0+ – CP0– = –100 mV	—	TBD	—	ns
Response Time: Mode 2, Vcm[1] = 1.5 V	CP0+ – CP0– = 100 mV	—	TBD	—	ns
	CP0+ – CP0– = –100 mV	—	TBD	—	ns
Response Time: Mode 3, Vcm[1] = 1.5 V	CP0+ – CP0– = 100 mV	—	TBD	—	ns
	CP0+ – CP0– = –100 mV	—	TBD	—	ns
Common-Mode Rejection Ratio		—	TBD	TBD	mV/V
Positive Hysteresis 1	CP0HYP1-0 = 00	—	TBD	TBD	mV
Positive Hysteresis 2	CP0HYP1-0 = 01	TBD	TBD	TBD	mV
Positive Hysteresis 3	CP0HYP1-0 = 10	TBD	TBD	TBD	mV
Positive Hysteresis 4	CP0HYP1-0 = 11	TBD	TBD	TBD	mV
Negative Hysteresis 1	CP0HYN1-0 = 00	—	TBD	TBD	mV
Negative Hysteresis 2	CP0HYN1-0 = 01	TBD	TBD	TBD	mV
Negative Hysteresis 3	CP0HYN1-0 = 10	TBD	TBD	TBD	mV
Negative Hysteresis 4	CP0HYN1-0 = 11	TBD	TBD	TBD	mV
Inverting or Non-Inverting Input Voltage Range		–0.25	—	V_{DD} + 0.25	V
Input Capacitance		—	TBD	—	pF
Input Bias Current		—	TBD	—	nA
Input Offset Voltage		TBD	—	TBD	mV
Input Impedance	High Quality Mode (CP1HIQE = ‘1’) Low Quality Mode (CP1HIQE = ‘0’)	—	TBD TBD	—	kΩ kΩ
Power Supply					
Power Supply Rejection[2]		—	TBD	TBD	mV/V
Power-up Time		—	TBD	—	µs
Power Consumption	High Quality Mode (CP1HIQE = ‘1’) Low Quality Mode (CP1HIQE = ‘0’)	—	TBD TBD	—	mA mA
Supply Current at DC	Mode 0	—	TBD	TBD	µA
	Mode 1	—	TBD	TBD	µA
	Mode 2	—	TBD	TBD	µA
	Mode 3	—	TBD	TBD	µA

Notes:
1. Vcm is the common-mode voltage on CP0+ and CP0–.
2. Guaranteed by design and/or characterization.

9. CIP-51 Microcontroller

The MCU system controller core is the CIP-51 microcontroller. The CIP-51 is fully compatible with the MCS-51™ instruction set. Standard 803x/805x assemblers and compilers can be used to develop software. The C8051F52x/C8051F53x family has a superset of all the peripherals included with a standard 8051. See Section "**1. System Overview**" on page **17** for more information about the available peripherals. The CIP-51 includes on-chip debug hardware which interfaces directly with the analog and digital subsystems, providing a complete data acquisition or control-system solution in a single integrated circuit.

The CIP-51 Microcontroller core implements the standard 8051 organization and peripherals as well as additional custom peripherals and functions to extend its capability (see Figure 9.1 for a block diagram). The CIP-51 core includes the following features:

- Fully Compatible with MCS-51 Instruction Set
- 25 MIPS Peak Throughput
- 256 Bytes of Internal RAM
- Extended Interrupt Handler
- Reset Input
- Power Management Modes
- Integrated Debug Logic
- Program and Data Memory Security

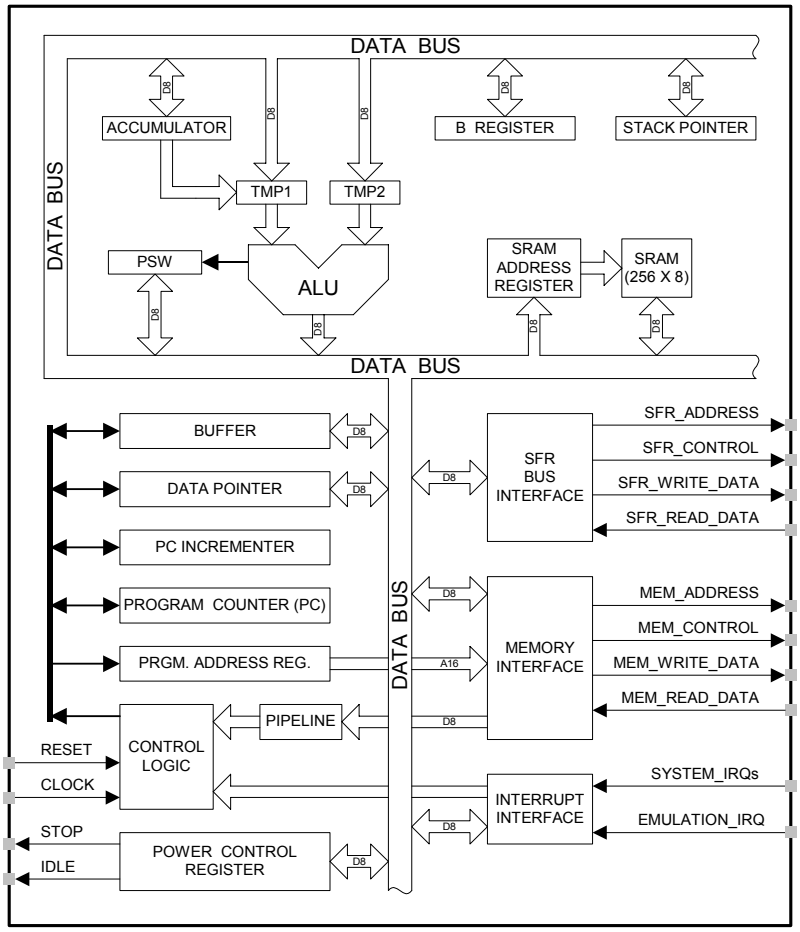


Figure 9.1. CIP-51 Block Diagram

Performance
The CIP-51 employs a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. In a standard 8051, all instructions except for MUL and DIV take 12 or 24 system clock cycles to execute, and usually have a maximum system clock of 12 MHz. By contrast, the CIP-51 core executes 70% of its instructions in one or two system clock cycles, with no instructions taking more than eight system clock cycles.

With the CIP-51's system clock running at 25 MHz, it has a peak throughput of 25 MIPS. The CIP-51 has a total of 109 instructions. The table below shows the total number of instructions that require each execution time.

Clocks to Execute	1	2	2/4	3	3/5	4	5	4/6	6	8
Number of Instructions	26	50	5	10	7	5	2	1	2	1

Programming and Debugging Support
In-system programming of the Flash program memory and communication with on-chip debug support logic is accomplished via the Silicon Labs 2-Wire (C2) interface. Note that the re-programmable Flash can also be read and written a single byte at a time by the application software using the MOVC and MOVX instructions. This feature allows program memory to be used for non-volatile data storage as well as updating program code under software control.

The on-chip debug support logic facilitates full speed in-circuit debugging, allowing the setting of hardware breakpoints, starting, stopping and single stepping through program execution (including interrupt service routines), examination of the program's call stack, and reading/writing the contents of registers and memory. This method of on-chip debugging is completely non-intrusive, requiring no RAM, Stack, timers, or other on-chip resources.

The CIP-51 is supported by development tools from Silicon Laboratories, Inc. and third party vendors. Silicon Laboratories provides an integrated development environment (IDE) including editor, evaluation compiler, assembler, debugger and programmer. The IDE's debugger and programmer interface to the CIP-51 via the on-chip debug logic to provide fast and efficient in-system device programming and debugging. Third party macro assemblers and C compilers are also available.

9.1. Instruction Set

The instruction set of the CIP-51 System Controller is fully compatible with the standard MCS-51™ instruction set. Standard 8051 development tools can be used to develop software for the CIP-51. All CIP-51 instructions are the binary and functional equivalent of their MCS-51™ counterparts, including opcodes, addressing modes and effect on PSW flags. However, instruction timing is different than that of the standard 8051.

9.1.1. Instruction and CPU Timing

In many 8051 implementations, a distinction is made between machine cycles and clock cycles, with machine cycles varying from 2 to 12 clock cycles in length. However, the CIP-51 implementation is based solely on clock cycle timing. All instruction timings are specified in terms of clock cycles.

Due to the pipelined architecture of the CIP-51, most instructions execute in the same number of clock cycles as there are program bytes in the instruction. Conditional branch instructions take two less clock cycles to complete when the branch is not taken as opposed to when the branch is taken. Table 9.1 is the CIP-51 Instruction Set Summary, which includes the mnemonic, number of bytes, and number of clock cycles for each instruction.

9.1.2. MOVX Instruction and Program Memory

The MOVX instruction is typically used to access data stored in XDATA memory space. In the CIP-51, the MOVX instruction can also be used to write or erase on-chip program memory space implemented as re-programmable Flash memory. The Flash access feature provides a mechanism for the CIP-51 to update program code and use the program memory space for non-volatile data storage. Refer to **Section "13. Flash Memory" on page 105** for further details.

Table 9.1. CIP-51 Instruction Set Summary[1]

Mnemonic	Description	Bytes	Clock Cycles
Arithmetic Operations			
ADD A, Rn	Add register to A	1	1
ADD A, direct	Add direct byte to A	2	2
ADD A, @Ri	Add indirect RAM to A	1	2
ADD A, #data	Add immediate to A	2	2
ADDC A, Rn	Add register to A with carry	1	1
ADDC A, direct	Add direct byte to A with carry	2	2
ADDC A, @Ri	Add indirect RAM to A with carry	1	2
ADDC A, #data	Add immediate to A with carry	2	2
SUBB A, Rn	Subtract register from A with borrow	1	1
SUBB A, direct	Subtract direct byte from A with borrow	2	2
SUBB A, @Ri	Subtract indirect RAM from A with borrow	1	2
SUBB A, #data	Subtract immediate from A with borrow	2	2
INC A	Increment A	1	1
INC Rn	Increment register	1	1
INC direct	Increment direct byte	2	2
INC @Ri	Increment indirect RAM	1	2
DEC A	Decrement A	1	1
DEC Rn	Decrement register	1	1
DEC direct	Decrement direct byte	2	2
DEC @Ri	Decrement indirect RAM	1	2
INC DPTR	Increment Data Pointer	1	1
MUL AB	Multiply A and B	1	4
DIV AB	Divide A by B	1	8
DA A	Decimal adjust A	1	1
Logical Operations			
ANL A, Rn	AND Register to A	1	1
ANL A, direct	AND direct byte to A	2	2
ANL A, @Ri	AND indirect RAM to A	1	2
ANL A, #data	AND immediate to A	2	2
ANL direct, A	AND A to direct byte	2	2
ANL direct, #data	AND immediate to direct byte	3	3
ORL A, Rn	OR Register to A	1	1
ORL A, direct	OR direct byte to A	2	2
ORL A, @Ri	OR indirect RAM to A	1	2
ORL A, #data	OR immediate to A	2	2
ORL direct, A	OR A to direct byte	2	2

Table 9.1. CIP-51 Instruction Set Summary[1] (Continued)

Mnemonic	Description	Bytes	Clock Cycles
ORL direct, #data	OR immediate to direct byte	3	3
XRL A, Rn	Exclusive-OR Register to A	1	1
XRL A, direct	Exclusive-OR direct byte to A	2	2
XRL A, @Ri	Exclusive-OR indirect RAM to A	1	2
XRL A, #data	Exclusive-OR immediate to A	2	2
XRL direct, A	Exclusive-OR A to direct byte	2	2
XRL direct, #data	Exclusive-OR immediate to direct byte	3	3
CLR A	Clear A	1	1
CPL A	Complement A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through Carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through Carry	1	1
SWAP A	Swap nibbles of A	1	1
Data Transfer			
MOV A, Rn	Move Register to A	1	1
MOV A, direct	Move direct byte to A	2	2
MOV A, @Ri	Move indirect RAM to A	1	2
MOV A, #data	Move immediate to A	2	2
MOV Rn, A	Move A to Register	1	1
MOV Rn, direct	Move direct byte to Register	2	2
MOV Rn, #data	Move immediate to Register	2	2
MOV direct, A	Move A to direct byte	2	2
MOV direct, Rn	Move Register to direct byte	2	2
MOV direct, direct	Move direct byte to direct byte	3	3
MOV direct, @Ri	Move indirect RAM to direct byte	2	2
MOV direct, #data	Move immediate to direct byte	3	3
MOV @Ri, A	Move A to indirect RAM	1	2
MOV @Ri, direct	Move direct byte to indirect RAM	2	2
MOV @Ri, #data	Move immediate to indirect RAM	2	2
MOV DPTR, #data16	Load DPTR with 16-bit constant	3	3
MOVC A, @A+DPTR	Move code byte relative DPTR to A	1	4 to 7[2]
MOVC A, @A+PC	Move code byte relative PC to A	1	4 to 7[2]
MOVX A, @Ri	Move external data (8-bit address) to A	1	3
MOVX @Ri, A	Move A to external data (8-bit address)	1	3
MOVX A, @DPTR	Move external data (16-bit address) to A	1	3
MOVX @DPTR, A	Move A to external data (16-bit address)	1	3
PUSH direct	Push direct byte onto stack	2	2
POP direct	Pop direct byte from stack	2	2
XCH A, Rn	Exchange Register with A	1	1
XCH A, direct	Exchange direct byte with A	2	2
XCH A, @Ri	Exchange indirect RAM with A	1	2
XCHD A, @Ri	Exchange low nibble of indirect RAM with A	1	2

Table 9.1. CIP-51 Instruction Set Summary[1] (Continued)

Mnemonic	Description	Bytes	Clock Cycles
Boolean Manipulation			
CLR C	Clear Carry	1	1
CLR bit	Clear direct bit	2	2
SETB C	Set Carry	1	1
SETB bit	Set direct bit	2	2
CPL C	Complement Carry	1	1
CPL bit	Complement direct bit	2	2
ANL C, bit	AND direct bit to Carry	2	2
ANL C, /bit	AND complement of direct bit to Carry	2	2
ORL C, bit	OR direct bit to carry	2	2
ORL C, /bit	OR complement of direct bit to Carry	2	2
MOV C, bit	Move direct bit to Carry	2	2
MOV bit, C	Move Carry to direct bit	2	2
JC rel	Jump if Carry is set	2	2/4
JNC rel	Jump if Carry is not set	2	2/4
JB bit, rel	Jump if direct bit is set	3	3/5
JNB bit, rel	Jump if direct bit is not set	3	3/5
JBC bit, rel	Jump if direct bit is set and clear bit	3	3/5
Program Branching			
ACALL addr11	Absolute subroutine call	2	4
LCALL addr16	Long subroutine call	3	5
RET	Return from subroutine	1	6
RETI	Return from interrupt	1	6
AJMP addr11	Absolute jump	2	4
LJMP addr16	Long jump	3	5
SJMP rel	Short jump (relative address)	2	4
JMP @A+DPTR	Jump indirect relative to DPTR	1	4
JZ rel	Jump if A equals zero	2	2/4
JNZ rel	Jump if A does not equal zero	2	2/4
CJNE A, direct, rel	Compare direct byte to A and jump if not equal	3	3/5
CJNE A, #data, rel	Compare immediate to A and jump if not equal	3	3/5
CJNE Rn, #data, rel	Compare immediate to Register and jump if not equal	3	3/5
CJNE @Ri, #data, rel	Compare immediate to indirect and jump if not equal	3	4/6
DJNZ Rn, rel	Decrement Register and jump if not zero	2	2/4
DJNZ direct, rel	Decrement direct byte and jump if not zero	3	3/5
NOP	No operation	1	1

Notes:

1. Assumes PFEN = 1 for all instruction timing.
2. MOVC instructions take 4 to 7 clock cycles depending on instruction alignment and the FLRT setting.

Notes on Registers, Operands and Addressing Modes:

Rn - Register R0–R7 of the currently selected register bank.

@Ri - Data RAM location addressed indirectly through R0 or R1.

rel - 8-bit, signed (two's complement) offset relative to the first byte of the following instruction. Used by SJMP and all conditional jumps.

direct - 8-bit internal data location's address. This could be a direct-access Data RAM location (0x00–0x7F) or an SFR (0x80–0xFF).

#data - 8-bit constant

#data16 - 16-bit constant

bit - Direct-accessed bit in Data RAM or SFR

addr11 - 11-bit destination address used by ACALL and AJMP. The destination must be within the same 2 kB page of program memory as the first byte of the following instruction.

addr16 - 16-bit destination address used by LCALL and LJMP. The destination may be anywhere within the 7680 bytes of program memory space.

There is one unused opcode (0xA5) that performs the same function as NOP.

9.2. Register Descriptions

Following are descriptions of SFRs related to the operation of the CIP-51 System Controller. Reserved bits should not be set to logic 1. Future product versions may use these bits to implement new features in which case the reset value of the bit will be logic 0, selecting the feature's default state. Detailed descriptions of the remaining SFRs are included in the sections of the datasheet associated with their corresponding system function.

SFR Definition 9.1. SP: Stack Pointer

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x81

Bits7–0: **SP**: Stack Pointer.
The Stack Pointer holds the location of the top of the stack. The stack pointer is incremented before every PUSH operation. The SP register defaults to 0x07 after reset.

SFR Definition 9.2. DPL: Data Pointer Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x82

Bits7–0: **DPL**: Data Pointer Low.
The DPL register is the low byte of the 16-bit DPTR. DPTR is used to access indirectly addressed XRAM and Flash memory.

SFR Definition 9.3. DPH: Data Pointer High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x83

Bits7–0: **DPH**: Data Pointer High.
The DPH register is the high byte of the 16-bit DPTR. DPTR is used to access indirectly addressed XRAM and Flash memory.

SFR Definition 9.4. PSW: Program Status Word

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	Reset Value
CY	AC	F0	RS1	RS0	OV	F1	PARITY	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0xD0

Bit7: **CY**: Carry Flag.
This bit is set when the last arithmetic operation resulted in a carry (addition) or a borrow (subtraction). It is cleared to 0 by all other arithmetic operations.

Bit6: **AC**: Auxiliary Carry Flag
This bit is set when the last arithmetic operation resulted in a carry into (addition) or a borrow from (subtraction) the high order nibble. It is cleared to 0 by all other arithmetic operations.

Bit5: **F0**: User Flag 0.
This is a bit-addressable, general purpose flag for use under software control.

Bits4–3: **RS1–RS0**: Register Bank Select.
These bits select which register bank is used during register accesses.

RS1	RS0	Register Bank	Address
0	0	0	0x00–0x07
0	1	1	0x08–0x0F
1	0	2	0x10–0x17
1	1	3	0x18–0x1F

Bit2: **OV**: Overflow Flag.
This bit is set to 1 under the following circumstances:
- An ADD, ADDC, or SUBB instruction causes a sign-change overflow.
- A MUL instruction results in an overflow (result is greater than 255).
- A DIV instruction causes a divide-by-zero condition.

The OV bit is cleared to 0 by the ADD, ADDC, SUBB, MUL, and DIV instructions in all other cases.

Bit1: **F1**: User Flag 1.
This is a bit-addressable, general purpose flag for use under software control.

Bit0: **PARITY**: Parity Flag.
This bit is set to 1 if the sum of the eight bits in the accumulator is odd and cleared if the sum is even.

SFR Definition 9.5. ACC: Accumulator

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0xE0

Bits7–0: **ACC**: Accumulator.
This register is the accumulator for arithmetic operations.

SFR Definition 9.6. B: B Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0xF0

Bits7–0: **B**: B Register.
This register serves as a second accumulator for certain arithmetic operations.

9.3. Power Management Modes

The CIP-51 core has two software programmable power management modes: Idle and Stop. Idle mode halts the CPU while leaving the peripherals and internal clocks active. In Stop mode, the CPU is halted, all interrupts and timers (except the Missing Clock Detector) are inactive, and the internal oscillator is stopped (analog peripherals remain in their selected states; the external oscillator is not affected). Since clocks are running in Idle mode, power consumption is dependent upon the system clock frequency and the number of peripherals left in active mode before entering Idle. Stop mode consumes the least power. SFR Definition 9.7 describes the Power Control Register (PCON) used to control the CIP-51's power management modes.

Although the CIP-51 has Idle and Stop modes built in (as with any standard 8051 architecture), power management of the entire MCU is better accomplished by enabling/disabling individual peripherals as needed. Each analog peripheral can be disabled when not in use and placed in low power mode. Digital peripherals, such as timers or serial buses, draw little power when they are not in use. Turning off the oscillators lowers power consumption considerably; however a reset is required to restart the MCU.

9.3.1. Idle Mode

Setting the Idle Mode Select bit (PCON.0) causes the CIP-51 to halt the CPU and enter Idle mode as soon as the instruction that sets the bit completes execution. All internal registers and memory maintain their original data. All analog and digital peripherals can remain active during Idle mode.

Idle mode is terminated when an enabled interrupt is asserted or a reset occurs. The assertion of an enabled interrupt will cause the Idle Mode Selection bit (PCON.0) to be cleared and the CPU to resume operation. The pending interrupt will be serviced and the next instruction to be executed after the return from interrupt (RETI) will be the instruction immediately following the one that set the Idle Mode Select bit. If Idle mode is terminated by an internal or external reset, the CIP-51 performs a normal reset sequence and begins program execution at address 0x0000.

If enabled, the Watchdog Timer (WDT) will eventually cause an internal watchdog reset and thereby terminate the Idle mode. This feature protects the system from an unintended permanent shutdown in the event of an inadvertent write to the PCON register. If this behavior is not desired, the WDT may be disabled by software prior to entering the Idle mode if the WDT was initially configured to allow this operation. This provides the opportunity for additional power savings, allowing the system to remain in the Idle mode indefinitely, waiting for an external stimulus to wake up the system.

9.3.2. Stop Mode

Setting the Stop Mode Select bit (PCON.1) causes the CIP-51 to enter Stop mode as soon as the instruction that sets the bit completes execution. In Stop mode the internal oscillator, CPU, and all digital peripherals are stopped; the state of the external oscillator circuit is not affected. Each analog peripheral (including the external oscillator circuit) may be shut down individually prior to entering Stop Mode. Stop mode can only be terminated by an internal or external reset. On reset, the CIP-51 performs the normal reset sequence and begins program execution at address 0x0000.

If enabled, the Missing Clock Detector will cause an internal reset and thereby terminate the Stop mode. The Missing Clock Detector should be disabled if the CPU is to be put to in STOP mode for longer than the MCD timeout period of 100 μs.

SFR Definition 9.7. PCON: Power Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	-	-	STOP	IDLE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x87

Bits7–2: **RESERVED**.

Bit1: **STOP**: STOP Mode Select.
Writing a '1' to this bit will place the CIP-51 into STOP mode. This bit will always read '0'.
1: CIP-51 forced into power-down mode. (Turns off internal oscillator).

Bit0: **IDLE**: IDLE Mode Select.
Writing a '1' to this bit will place the CIP-51 into IDLE mode. This bit will always read '0'.
1: CIP-51 forced into IDLE mode. (Shuts off clock to CPU, but clock to Timers, Interrupts, and all peripherals remain active.)

10. Memory Organization and SFRs

The memory organization of the C8051F52x/F53x is similar to that of a standard 8051. There are two separate memory spaces: program memory and data memory. Program and data memory share the same address space but are accessed via different instruction types. The memory map is shown in Figure 10.1.

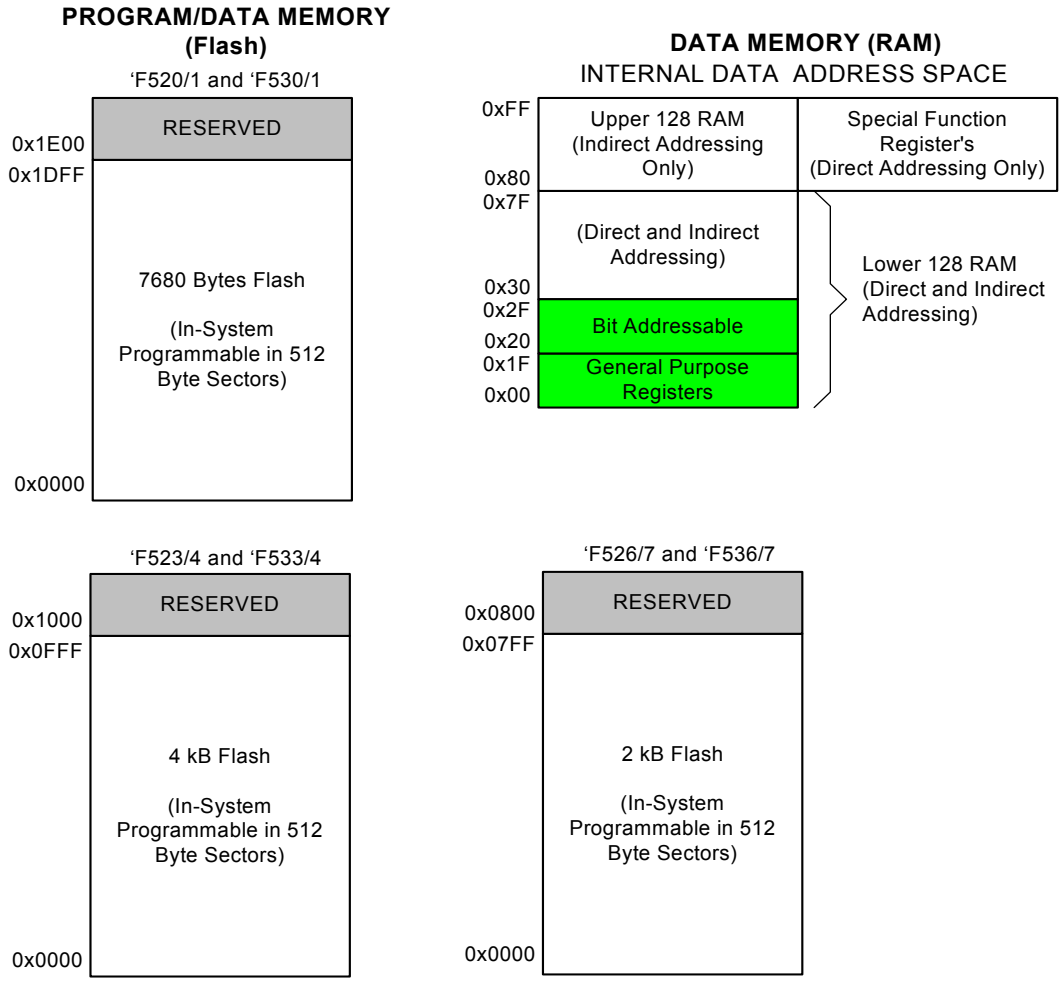


Figure 10.1. Memory Map

10.1. Program Memory

The CIP-51 core has a 64k-byte program memory space. The C8051F520/1 and 'F530/1 implement 8 kB of this program memory space as in-system, re-programmable Flash memory, organized in a contiguous block from addresses 0x0000 to 0x1FFF. Addresses above 0x1DFF are reserved on the 8 kB devices. The C8051F523/4 and 'F533/4 implement 4 kB of Flash from addresses 0x0000 to 0x0FFF.The C8051F526/7 and 'F536/7 implement 2 kB of Flash from addresses 0x0000 to 0x07FF.

Program memory is normally assumed to be read-only. However, the C8051F52x/F53x can write to program memory by setting the Program Store Write Enable bit (PSCTL.0) and using the MOVX write instruction. This feature provides a mechanism for updates to program code and use of the program memory space for non-volatile data storage. Refer to **Section "13. Flash Memory" on page 105** for further details.

10.2. Data Memory

The C8051F52x/F53x includes 256 bytes of internal RAM mapped into the data memory space from 0x00 through 0xFF. The lower 128 bytes of data memory are used for general purpose registers and scratch pad memory. Either direct or indirect addressing may be used to access the lower 128 bytes of data memory. Locations 0x00 through 0x1F are addressable as four banks of general purpose registers, each bank consisting of eight byte-wide registers. The next 16 bytes, locations 0x20 through 0x2F, may either be addressed as bytes or as 128 bit locations accessible with the direct addressing mode.

The upper 128 bytes of data memory are accessible only by indirect addressing. This region occupies the same address space as the Special Function Registers (SFRs) but is physically separate from the SFR space. The addressing mode used by an instruction when accessing locations above 0x7F determines whether the CPU accesses the upper 128 bytes of data memory space or the SFRs. Instructions that use direct addressing will access the SFR space. Instructions using indirect addressing above 0x7F access the upper 128 bytes of data memory. Figure 10.1 illustrates the data memory organization of the C8051F52x/C8051F53x.

10.3. General Purpose Registers

The lower 32 bytes of data memory (locations 0x00 through 0x1F) may be addressed as four banks of general-purpose registers. Each bank consists of eight byte-wide registers designated R0 through R7. Only one of these banks may be enabled at a time. Two bits in the program status word, RS0 (PSW.3) and RS1 (PSW.4), select the active register bank (see description of the PSW in SFR Definition 9.4. PSW: Program Status Word). This allows fast context switching when entering subroutines and interrupt service routines. Indirect addressing modes use registers R0 and R1 as index registers.

10.4. Bit Addressable Locations

In addition to direct access to data memory organized as bytes, the sixteen data memory locations at 0x20 through 0x2F are also accessible as 128 individually addressable bits. Each bit has a bit address from 0x00 to 0x7F. Bit 0 of the byte at 0x20 has bit address 0x00 while bit 7 of the byte at 0x20 has bit address 0x07. Bit 7 of the byte at 0x2F has bit address 0x7F. A bit access is distinguished from a full byte access by the type of instruction used (bit source or destination operands as opposed to a byte source or destination).

The MCS-51™ assembly language allows an alternate notation for bit addressing of the form XX.B where XX is the byte address and B is the bit position within the byte. For example, the instruction:

```
MOV    C, 22.3h
```

moves the Boolean value at 0x13 (bit 3 of the byte at location 0x22) into the Carry flag.

10.5. Stack

A programmer's stack can be located anywhere in the 256-byte data memory. The stack area is designated using the Stack Pointer (SP, 0x81) SFR. The SP will point to the last location used. The next value pushed on the stack is placed at SP+1 and then SP is incremented. A reset initializes the stack pointer to location 0x07. Therefore, the first value pushed on the stack is placed at location 0x08, which is also the first register (R0) of register bank 1. Thus, if more than one register bank is to be used, the SP should be initialized to a location in the data memory not being used for data storage. The stack depth can extend up to 256 bytes.

10.6. Special Function Registers

The direct-access data memory locations from 0x80 to 0xFF constitute the special function registers (SFRs). The SFRs provide control and data exchange with the CIP-51's resources and peripherals. The CIP-51 duplicates the SFRs found in a typical 8051 implementation as well as implementing additional SFRs used to configure and access the sub-systems unique to the MCU. This allows the addition of new functionality while retaining compatibility with the MCS-51™ instruction set. Table 10.1 lists the SFRs implemented in the CIP-51 System Controller.

The SFR registers are accessed anytime the direct addressing mode is used to access memory locations from 0x80 to 0xFF. SFRs with addresses ending in 0x0 or 0x8 (e.g. P0, TCON, IE, etc.) are bit-addressable as well as byte-addressable. All other SFRs are byte-addressable only. Unoccupied addresses in the SFR space are reserved for future use. Accessing these areas will have an indeterminate effect and should be avoided. Refer to the corresponding pages of the data sheet, as indicated in Table 10.2, for a detailed description of each register.

Table 10.1. Special Function Register (SFR) Memory Map

F8	SPI0CN	PCA0L	PCA0H	PCA0CPL0	PCA0CPH0			VDDMON
F0	B	P0MDIN	P1MDIN				EIP1	
E8	ADC0CN	PCA0CPL1	PCA0CPH1	PCA0CPL2	PCA0CPH2			RSTSRC
E0	ACC	XBR0	XBR1		IT01CF		EIE1	
D8	PCA0CN	PCA0MD	PCA0CPM0	PCA0CPM1	PCA0CPM2			
D0	PSW	REF0CN			P0SKIP	P1SKIP		P0MAT
C8	TMR2CN	REG0CN	TMR2RLL	TMR2RLH	TMR2L	TMR2H		P1MAT
C0				ADC0GTL	ADC0GTH	ADC0LTL	ADC0LTH	P0MASK
B8	IP		ADC0TK	ADC0MX	ADC0CF	ADC0L	ADC0	P1MASK
B0	OSCIFIN	OSCXCN	OSCICN	OSCICL				FLKEY
A8	IE	CLKSEL						
A0	—	SPI0CFG	SPI0CKR	SPI0DAT	P0MDOUT	P1MDOUT		
98	SCON0	SBUF0		CPT0CN		CPT0MD		CPT0MX
90	P1		LINADDR	LINDATA		LINCF		
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	PSCTL
80	P0	SP	DPL	DPH				PCON
		1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

Table 10.2. Special Function Registers

SFRs are listed in alphabetical order. All undefined SFR locations are reserved

Register	Address	Description	Page
ACC	0xE0	Accumulator	81
ADC0CF	0xBC	ADC0 Configuration	50
ADC0CN	0xE8	ADC0 Control	52
ADC0H	0xBE	ADC0	51
ADC0L	0xBD	ADC0	51
ADC0GTH	0xC4	ADC0 Greater-Than Data High Byte	54
ADC0GTL	0xC3	ADC0 Greater-Than Data Low Byte	54
ADC0LTH	0xC6	ADC0 Less-Than Data High Byte	55
ADC0LTL	0xC5	ADC0 Less-Than Data Low Byte	55
ADC0MX	0xBB	ADC0 Channel Select	49
ADC0TK	0xBA	ADC0 Tracking Mode Select	53
B	0xF0	B Register	81
CKCON	0x8E	Clock Control	185
CLKSEL	0xA9	Clock Select	139
CPT0CN	0x9B	Comparator0 Control	69
CPT0MD	0x9D	Comparator0 Mode Selection	71
CPT0MX	0x9F	Comparator0 MUX Selection	70
DPH	0x83	Data Pointer High	79
DPL	0x82	Data Pointer Low	79
EIE1	0xE6	Extended Interrupt Enable 1	93
EIP1	0xF6	Extended Interrupt Priority 1	94
FLKEY	0xB7	Flash Lock and Key	113
IE	0xA8	Interrupt Enable	91
IP	0xB8	Interrupt Priority	92
IT01CF	0xE4	INT0/INT1 Configuration	96
LINADDR	0x92	LIN indirect address pointer	151
LINCF	0x95	LIN master-slave and automatic baud rate selection	151
LINDATA	0x93	LIN indirect data buffer	151
OSCICL	0xB3	Internal Oscillator Calibration	134
OSCICN	0xB2	Internal Oscillator Control	133
OSCXCN	0xB1	External Oscillator Control	138
P0	0x80	Port 0 Latch	124
P0MASK	0xC7	Port 0 Mask	126
P0MAT	0xD7	Port 0 Match	126
P0MDIN	0xF1	Port 0 Input Mode Configuration	124
P0MDOUT	0xA4	Port 0 Output Mode Configuration	125
P0SKIP	0xD4	Port 0 Skip	125
P1	0x90	Port 1 Latch	127

Table 10.2. Special Function Registers (Continued)

SFRs are listed in alphabetical order. All undefined SFR locations are reserved

Register	Address	Description	Page
P1MASK	0xBF	Port 1 Mask	129
P1MAT	0xCF	Port 1 Match	129
P1MDIN	0xF2	Port 1 Input Mode Configuration	127
P1MDOUT	0xA5	Port 1 Output Mode Configuration	128
P1SKIP	0xD5	Port 1 Skip	128
PCA0CN	0xD8	PCA Control	205
PCA0CPH0	0xFC	PCA Capture 0 High	208
PCA0CPH1	0xEA	PCA Capture 1 High	208
PCA0CPH2	0xEC	PCA Capture 2 High	208
PCA0CPL0	0xFB	PCA Capture 0 Low	208
PCA0CPL1	0xE9	PCA Capture 1 Low	208
PCA0CPL2	0xEB	PCA Capture 2 Low	208
PCA0CPM0	0xDA	PCA Module 0 Mode	207
PCA0CPM1	0xDB	PCA Module 1 Mode	207
PCA0CPM2	0xDC	PCA Module 2 Mode	207
PCA0H	0xFA	PCA Counter High	208
PCA0L	0xF9	PCA Counter Low	208
PCA0MD	0xD9	PCA Mode	206
PCON	0x87	Power Control	82
PSCTL	0x8F	Program Store R/W Control	113
PSW	0xD0	Program Status Word	80
REF0CN	0xD1	Voltage Reference Control	62
REG0CN	0xC9	Voltage Regulator Control	66
RSTSRC	0xEF	Reset Source Configuration/Status	102
SBUF0	0x99	UART0 Data Buffer	147
SCON0	0x98	UART0 Control	146
SP	0x81	Stack Pointer	78
SPI0CFG	0xA1	SPI Configuration	171
SPI0CKR	0xA2	SPI Clock Rate Control	173
SPI0CN	0xF8	SPI Control	172
SPI0DAT	0xA3	SPI Data	174
TCON	0x88	Timer/Counter Control	183
TH0	0x8C	Timer/Counter 0 High	186
TH1	0x8D	Timer/Counter 1 High	186
TL0	0x8A	Timer/Counter 0 Low	186
TL1	0x8B	Timer/Counter 1 Low	186
TMOD	0x89	Timer/Counter Mode	184
TMR2CN	0xC8	Timer/Counter 2 Control	190

Table 10.2. Special Function Registers (Continued)

SFRs are listed in alphabetical order. All undefined SFR locations are reserved

Register	Address	Description	Page
TMR2H	0xCD	Timer/Counter 2 High	191
TMR2L	0xCC	Timer/Counter 2 Low	191
TMR2RLH	0xCB	Timer/Counter 2 Reload High	191
TMR2RLL	0xCA	Timer/Counter 2 Reload Low	191
VDDMON	0xFF	V_{DD} Monitor Control	100
XBR0	0xE1	Port I/O Crossbar Control 0	122
XBR1	0xE2	Port I/O Crossbar Control 1	123

11. Interrupt Handler

The C8051F52x/F53x family includes an extended interrupt system with two selectable priority levels. The allocation of interrupt sources between on-chip peripherals and external input pins varies according to the specific version of the device. Each interrupt source has one or more associated interrupt-pending flag(s) located in an SFR. When a peripheral or external source meets a valid interrupt condition, the associated interrupt-pending flag is set to logic 1.

If interrupts are enabled for the source, an interrupt request is generated when the interrupt-pending flag is set. As soon as execution of the current instruction is complete, the CPU generates an LCALL to a predetermined address to begin execution of an interrupt service routine (ISR). Each ISR must end with an RETI instruction, which returns program execution to the next instruction that would have been executed if the interrupt request had not occurred. If interrupts are not enabled, the interrupt-pending flag is ignored by the hardware and program execution continues as normal. (The interrupt-pending flag is set to logic 1 regardless of the interrupt's enable/disable state.)

Each interrupt source can be individually enabled or disabled through the use of an associated interrupt enable bit in the Interrupt Enable and Extended Interrupt Enable SFRs. However, interrupts must first be globally enabled by setting the EA bit (IE.7) to logic 1 before the individual interrupt enables are recognized. Setting the EA bit to logic 0 disables all interrupt sources regardless of the individual interrupt-enable settings. Note that interrupts which occur when the EA bit is set to logic 0 will be held in a pending state, and will not be serviced until the EA bit is set back to logic 1.

Some interrupt-pending flags are automatically cleared by the hardware when the CPU vectors to the ISR. However, most are not cleared by the hardware and must be cleared by software before returning from the ISR. If an interrupt-pending flag remains set after the CPU completes the return-from-interrupt (RETI) instruction, a new interrupt request will be generated immediately and the CPU will re-enter the ISR after the completion of the next instruction.

11.1. MCU Interrupt Sources and Vectors

The MCUs support 15 interrupt sources. Software can simulate an interrupt by setting any interrupt-pending flag to logic 1. If interrupts are enabled for the flag, an interrupt request will be generated and the CPU will vector to the ISR address associated with the interrupt-pending flag. MCU interrupt sources, associated vector addresses, priority order, and control bits are summarized in Table 11.1 on page 90. Refer to the data sheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

11.2. Interrupt Priorities

Each interrupt source can be individually programmed to one of two priority levels: low or high. A low priority interrupt service routine can be preempted by a high priority interrupt. A high priority interrupt cannot be preempted. Each interrupt has an associated interrupt priority bit in an SFR (IP or EIP1) used to configure its priority level. Low priority is the default. If two interrupts are recognized simultaneously, the interrupt with the higher priority is serviced first. If both interrupts have the same priority level, a fixed priority order is used to arbitrate, given in Table 11.1.

11.3. Interrupt Latency

Interrupt response time depends on the state of the CPU when the interrupt occurs. Pending interrupts are sampled and priority decoded each system clock cycle. Therefore, the fastest possible response time is 7 system clock cycles: 1 clock cycle to detect the interrupt, 1 clock cycle to execute a single instruction, and 5 clock cycles to complete the LCALL to the ISR. If an interrupt is pending when a RETI is executed, a single instruction is executed before an LCALL is made to service the pending interrupt. Therefore, the maximum response time for an interrupt (when no other interrupt is currently being serviced or the new interrupt is of greater priority) occurs when the CPU is performing an RETI instruction followed by a DIV as the next

instruction. In this case, the response time is 19 system clock cycles: 1 clock cycle to detect the interrupt, 5 clock cycles to execute the RETI, 8 clock cycles to complete the DIV instruction and 5 clock cycles to execute the LCALL to the ISR. If the CPU is executing an ISR for an interrupt with equal or higher priority, the new interrupt will not be serviced until the current ISR completes, including the RETI and following instruction.

Table 11.1. Interrupt Summary

Interrupt Source	Interrupt Vector	Priority Order	Pending Flag	Bit addressable?	Cleared by HW?	Enable Flag	Priority Control
Reset	0x0000	Top	None	N/A	N/A	Always Enabled	Always Highest
External Interrupt 0(/INT0)	0x0003	0	IE0 (TCON.1)	Y	Y	EX0 (IE.0)	PX0 (IP.0)
Timer 0 Overflow	0x000B	1	TF0 (TCON.5)	Y	Y	ET0 (IE.1)	PT0 (IP.1)
External Interrupt 1(/INT1)	0x0013	2	IE1 (TCON.3)	Y	Y	EX1 (IE.2)	PX1 (IP.2)
Timer 1 Overflow	0x001B	3	TF1 (TCON.7)	Y	Y	ET1 (IE.3)	PT1 (IP.3)
UART	0x0023	4	RI0 (SCON0.0) TI0 (SCON0.1)	Y	N	ES0 (IE.4)	PS0 (IP.4)
Timer 2 Overflow	0x002B	5	TF2H (TMR2CN.7) TF2L (TMR2CN.6)	Y	N	ET2 (IE.5)	PT2 (IP.5)
SPI0	0x0033	6	SPIF (SPI0CN.7) WCOL (SPI0CN.6) MODF (SPI0CN.5) RXOVRN (SPI0CN.4)	Y	N	ESPI0 (IE.6)	PSPI0 (IP.6)
ADC0 Window Comparator	0x003B	7	AD0WINT (ADC0CN.3)	Y	N	EWADC0 (EIE1.0)	PWADC0 (EIP1.0)
ADC0 End of Conversion	0x0043	8	AD0INT (ADC0CN.5)	Y	N	EADC0 (EIE1.1)	PADC0 (EIP1.1)
Programmable Counter Array	0x004B	9	CF (PCA0CN.7) CCFn (PCA0CN.n)	Y	N	EPCA0 (EIE1.2)	PPCA0 (EIP1.2)
Comparator Falling Edge	0x0053	10	CP0FIF (CPT0CN.4) CP0RIF (CPT0CN.5)	N	N	ECPF (EIE1.3)	PCPF (EIP1.3)
Comparator Rising Edge	0x005B	11	CP0FIF (CPT0CN.4) CP0RIF (CPT0CN.5)	N	N	ECPR (EIE1.4)	PCPR (EIP1.4)
LIN Interrupt	0x0063	12	LININT (LINST.3)	N	N*	ELIN (EIE1.5)	PLIN (EIP1.5)
Voltage Regulator Dropout	0x006B	13	N/A	N/A	N/A	EREG0 (EIE1.6)	PREG0 (EIP1.6)
Port Match	0x0073	14	N/A	N/A	N/A	EMAT (EIE1.7)	PMAT (EIP1.7)
***Note:** To clear LININT requires the application to set the RSTINT bit (LINCTRL.3)							

11.4. Interrupt Register Descriptions

The SFRs used to enable the interrupt sources and set their priority level are described below. Refer to the data sheet section associated with a particular on-chip peripheral for information regarding valid interrupt conditions for the peripheral and the behavior of its interrupt-pending flag(s).

SFR Definition 11.1. IE: Interrupt Enable

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
EA	ESPI0	ET2	ES0	ET1	EX1	ET0	EX0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0xA8

Bit7: **EA**: Global Interrupt Enable.
This bit globally enables/disables all interrupts. It overrides the individual interrupt mask settings.
0: Disable all interrupt sources.
1: Enable each interrupt according to its individual mask setting.

Bit6: **ESPI0**: Enable Serial Peripheral Interface (SPI0) Interrupt.
This bit sets the masking of the SPI0 interrupts.
0: Disable all SPI0 interrupts.
1: Enable interrupt requests generated by SPI0.

Bit5: **ET2**: Enable Timer 2 Interrupt.
This bit sets the masking of the Timer 2 interrupt.
0: Disable Timer 2 interrupt.
1: Enable interrupt requests generated by the TF2L or TF2H flags.

Bit4: **ES0**: Enable UART0 Interrupt.
This bit sets the masking of the UART0 interrupt.
0: Disable UART0 interrupt.
1: Enable UART0 interrupt.

Bit3: **ET1**: Enable Timer 1 Interrupt.
This bit sets the masking of the Timer 1 interrupt.
0: Disable all Timer 1 interrupt.
1: Enable interrupt requests generated by the TF1 flag.

Bit2: **EX1**: Enable External Interrupt 1.
This bit sets the masking of the external interrupt 1.
0: Disable external interrupt 1.
1: Enable extern interrupt 1 requests.

Bit1: **ET0**: Enable Timer 0 Interrupt.
This bit sets the masking of the Timer 0 interrupt.
0: Disable all Timer 0 interrupt.
1: Enable interrupt requests generated by the TF0 flag.

Bit0: **EX0**: Enable External Interrupt 0.
This bit sets the masking of the external interrupt 0.
0: Disable external interrupt 0.
1: Enable extern interrupt 0 requests.

SFR Definition 11.2. IP: Interrupt Priority

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	PSPI0	PT2	PS0	PT1	PX1	PT0	PX0	10000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0xB8

Bit7: **UNUSED**. Read = 1, Write = don't care.
Bit6: **PSPI0**: Serial Peripheral Interface (SPI0) Interrupt Priority Control.
This bit sets the priority of the SPI0 interrupt.
0: SPI0 interrupt set to low priority level.
1: SPI0 interrupt set to high priority level.
Bit5: **PT2**: Timer 2 Interrupt Priority Control.
This bit sets the priority of the Timer 2 interrupt.
0: Timer 2 interrupt set to low priority level.
1: Timer 2 interrupt set to high priority level.
Bit4: **PS0**: UART0 Interrupt Priority Control.
This bit sets the priority of the UART0 interrupt.
0: UART0 interrupt set to low priority level.
1: UART0 interrupt set to high priority level.
Bit3: **PT1**: Timer 1 Interrupt Priority Control.
This bit sets the priority of the Timer 1 interrupt.
0: Timer 1 interrupt set to low priority level.
1: Timer 1 interrupt set to high priority level.
Bit2: **PX1**: External Interrupt 0 Priority Control.
This bit sets the priority of the external interrupt 0.
0: Int 0 interrupt set to low priority level.
1: Int 0 interrupt set to high priority level.
Bit1: **PT0**: Timer 0 Interrupt Priority Control.
This bit sets the priority of the Timer 0 interrupt.
0: Timer 0 interrupt set to low priority level.
1: Timer 0 interrupt set to high priority level.
Bit0: **PX0**: External Interrupt 0 Priority Control.
This bit sets the priority of the external interrupt 0.
0: Int 0 interrupt set to low priority level.
1: Int 0 interrupt set to high priority level.

SFR Definition 11.3. EIE1: Extended Interrupt Enable 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
EMAT	EREG0	ELIN	ECPR	ECPF	EPCA0	EADC0	EWADC0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xE6

Bit7: **EMAT**:Enable Port Match Interrupt.
This bit sets the masking of the Port Match interrupt.
0: Disable the Port Match interrupt.
1: Enable the Port Match interrupt.

Bit6: **EREG0**: Enable Voltage Regulator Interrupt.
This bit sets the masking of the Voltage Regulator Dropout interrupt.
0: Disable the Voltage Regulator Dropout interrupt.
1: Enable the Voltage Regulator Dropout interrupt.

Bit5: **ELIN**: Enable LIN Interrupt.
This bit sets the masking of the LIN interrupt.
0: Disable LIN interrupts.
1: Enable LIN interrupt requests.

Bit4: **ECPR**: Enable Comparator 0 Rising Edge Interrupt
This bit sets the masking of the CP0 Rising Edge interrupt.
0: Disable CP0 Rising Edge Interrupt.
1: Enable CP0 Rising Edge Interrupt.

Bit3: **ECPF**: Enable Comparator 0 Falling Edge Interrupt
This bit sets the masking of the CP0 Falling Edge interrupt.
0: Disable CP0 Falling Edge Interrupt.
1: Enable CP0 Falling Edge Interrupt.

Bit2: **EPCA0**: Enable Programmable Counter Array (PCA0) Interrupt.
This bit sets the masking of the PCA0 interrupts.
0: Disable all PCA0 interrupts.
1: Enable interrupt requests generated by PCA0.

Bit1: **EADC0**: Enable ADC0 Conversion Complete Interrupt.
This bit sets the masking of the ADC0 Conversion Complete interrupt.
0: Disable ADC0 Conversion Complete interrupt.
1: Enable interrupt requests generated by the AD0INT flag.

Bit0: **EWADC0**: Enable ADC0 Window Comparison Interrupt.
This bit sets the masking of the ADC0 Window Comparison interrupt.
0: Disable ADC0 Window Comparison interrupt.
1: Enable interrupt requests generated by the AD0WINT flag.

SFR Definition 11.4. EIP1: Extended Interrupt Priority 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
PMAT	PREG0	PLIN	PCPR	PCPF	PPAC0	PREG0	PWADC0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xF6

Bit7: **PMAT**. Port Match Interrupt Priority Control.
This bit sets the priority of the Port Match interrupt.
0: Port Match interrupt set to low priority level.
1: Port Match interrupt set to high priority level.

Bit6: **PREG0**: Voltage Regulator Interrupt Priority Control.
This bit sets the priority of the Voltage Regulator interrupt.
0: Voltage Regulator interrupt set to low priority level.
1: Voltage Regulator interrupt set to high priority level.

Bit5: **PLIN**: LIN Interrupt Priority Control.
This bit sets the priority of the CP0 interrupt.
0: LIN interrupt set to low priority level.
1: LIN interrupt set to high priority level.

Bit4: **PCPR**: Comparator Rising Edge Interrupt Priority Control.
This bit sets the priority of the Rising Edge Comparator interrupt.
0: Comparator interrupt set to low priority level.
1: Comparator interrupt set to high priority level.

Bit3: **PCPF**: Comparator falling Edge Interrupt Priority Control.
This bit sets the priority of the Falling Edge Comparator interrupt.
0: Comparator interrupt set to low priority level.
1: Comparator interrupt set to high priority level.

Bit2: **PPAC0**: Programmable Counter Array (PCA0) Interrupt Priority Control.
This bit sets the priority of the PCA0 interrupt.
0: PCA0 interrupt set to low priority level.
1: PCA0 interrupt set to high priority level.

Bit1: **PREG0**: ADC0 Conversion Complete Interrupt Priority Control.
This bit sets the priority of the ADC0 Conversion Complete interrupt.
0: ADC0 Conversion Complete interrupt set to low priority level.
1: ADC0 Conversion Complete interrupt set to high priority level.

Bit0: **PWADC0**: ADC0 Window Comparison Interrupt Priority Control.
This bit sets the priority of the ADC0 Window Comparison interrupt.
0: ADC0 Window Comparison interrupt set to low priority level.
1: ADC0 Window Comparison interrupt set to high priority level.

11.5. External Interrupts

The /INT0 and /INT1 external interrupt sources are configurable as active high or low, edge or level sensitive. The IN0PL (/INT0 Polarity) and IN1PL (/INT1 Polarity) bits in the IT01CF register select active high or active low; the IT0 and IT1 bits in TCON (**Section "19.1. Timer 0 and Timer 1" on page 179**) select level or edge sensitive. The table below lists the possible configurations.

IT0	IN0PL	/INT0 Interrupt
1	0	Active low, edge sensitive
1	1	Active high, edge sensitive
0	0	Active low, level sensitive
0	1	Active high, level sensitive

IT1	IN1PL	/INT1 Interrupt
1	0	Active low, edge sensitive
1	1	Active high, edge sensitive
0	0	Active low, level sensitive
0	1	Active high, level sensitive

/INT0 and /INT1 are assigned to Port pins as defined in the IT01CF register (see SFR Definition 11.5). Note that /INT0 and /INT0 Port pin assignments are independent of any Crossbar assignments. /INT0 and /INT1 will monitor their assigned Port pins without disturbing the peripheral that was assigned the Port pin via the Crossbar. To assign a Port pin only to /INT0 and/or /INT1, configure the Crossbar to skip the selected pin(s). This is accomplished by setting the associated bit in register XBR0 (see **Section "14.1. Priority Crossbar Decoder" on page 117** for complete details on configuring the Crossbar).

IE0 (TCON.1) and IE1 (TCON.3) serve as the interrupt-pending flags for the /INT0 and /INT1 external interrupts, respectively. If an /INT0 or /INT1 external interrupt is configured as edge-sensitive, the corresponding interrupt-pending flag is automatically cleared by the hardware when the CPU vectors to the ISR. When configured as level sensitive, the interrupt-pending flag remains logic 1 while the input is active as defined by the corresponding polarity bit (IN0PL or IN1PL); the flag remains logic 0 while the input is inactive. The external interrupt source must hold the input active until the interrupt request is recognized. It must then deactivate the interrupt request before execution of the ISR completes or another interrupt request will be generated.

SFR Definition 11.5. IT01CF: INT0/INT1 Configuration

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
IN1PL	IN1SL2	IN1SL1	IN1SL0	IN0PL	IN0SL2	IN0SL1	IN0SL0	00000001
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xE4

Note: Refer to SFR Definition 19.1. “TCON: Timer Control” on page 183 for INT0/1 edge- or level-sensitive interrupt selection.

Bit 7: **IN1PL**: /INT1 Polarity
0: /INT1 input is active low.
1: /INT1 input is active high.

Bits 6–4: **IN1SL2–0**: /INT1 Port Pin Selection Bits
These bits select which Port pin is assigned to /INT1. Note that this pin assignment is independent of the Crossbar; /INT1 will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin (accomplished by setting to ‘1’ the corresponding bit in register P0SKIP).

IN1SL2-0	/INT1 Port Pin
000	P0.0
001	P0.1
010	P0.2
011	P0.3
100	P0.4
101	P0.5
110	P0.6*
111	P0.7*
***Note:** Available in the C80151F53x parts.	

Bit 3: **IN0PL**: /INT0 Polarity
0: /INT0 interrupt is active low.
1: /INT0 interrupt is active high.

Bits 2–0: **INT0SL2–0**: /INT0 Port Pin Selection Bits
These bits select which Port pin is assigned to /INT0. Note that this pin assignment is independent of the Crossbar. /INT0 will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin (accomplished by setting to ‘1’ the corresponding bit in register P0SKIP).

IN0SL2-0	/INT0 Port Pin
000	P0.0
001	P0.1
010	P0.2
011	P0.3
100	P0.4
101	P0.5
110	P0.6*
111	P0.7*
***Note:** Available in the C80151F53x parts.	

12. Reset Sources

Reset circuitry allows the controller to be easily placed in a predefined default condition. On entry to this reset state, the following occur:

- CIP-51 halts program execution
- Special Function Registers (SFRs) are initialized to their defined reset values
- External Port pins are forced to a known state
- Interrupts and timers are disabled.

All SFRs are reset to the predefined values noted in the SFR detailed descriptions. The contents of internal data memory are unaffected during a reset; any previously stored data is preserved. However, since the stack pointer SFR is reset, the stack is effectively lost, even though the data on the stack is not altered.

The Port I/O latches are reset to 0xFF (all logic ones) in open-drain mode. Weak pullups are enabled during and after the reset. For V_{DD} Monitor and power-on resets, the $\overline{RST}$ pin is driven low until the device exits the reset state.

On exit from the reset state, the program counter (PC) is reset, and the system clock defaults to the internal oscillator. Refer to **Section "15. Oscillators" on page 131** for information on selecting and configuring the system clock source. The Watchdog Timer is enabled with the system clock divided by 12 as its clock source (**Section "20.3. Watchdog Timer Mode" on page 201** details the use of the Watchdog Timer). Program execution begins at location 0x0000.

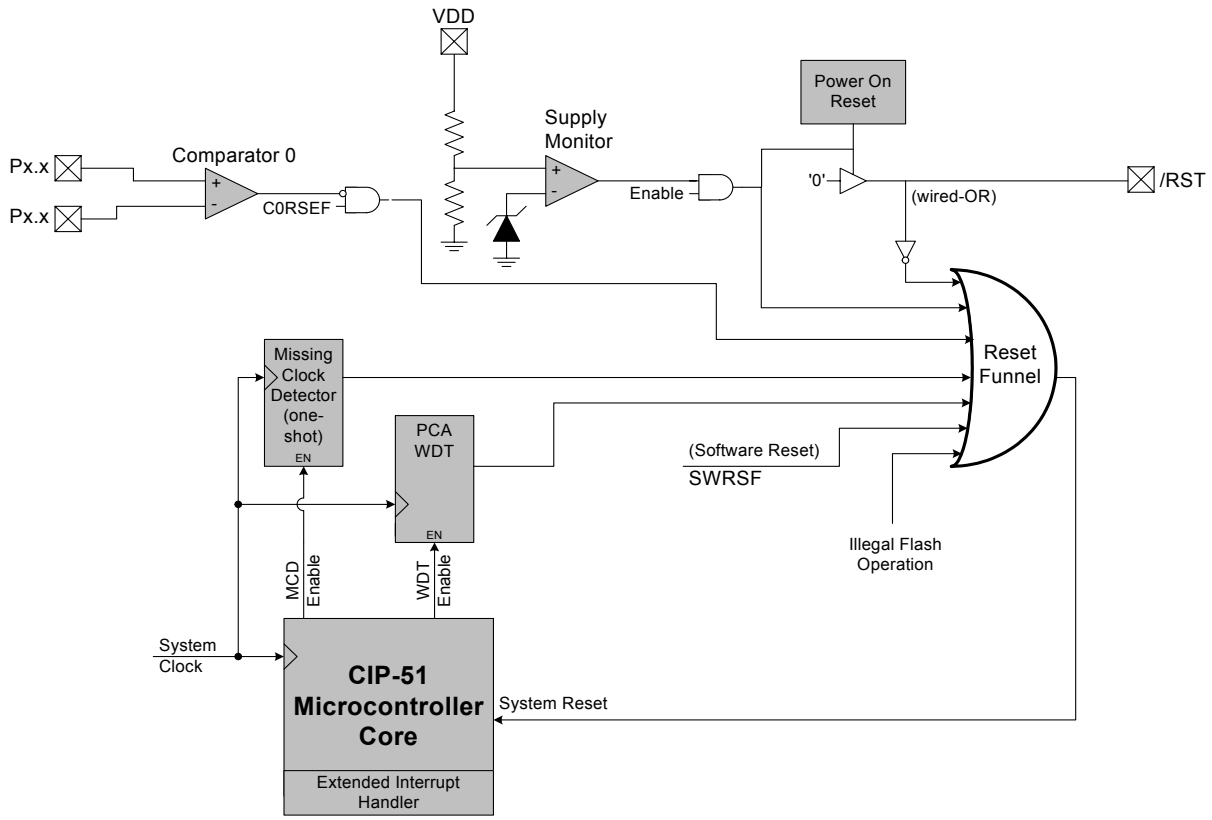


Figure 12.1. Reset Sources

12.1. Power-On Reset

During power-up, the device is held in a reset state and the $\overline{\text{RST}}$ pin is driven low until V_{DD} settles above V_{RST}. An additional delay occurs before the device is released from reset; the delay decreases as the V_{DD} ramp time increases (V_{DD} ramp time is defined as how fast V_{DD} ramps from 0 V to V_{RST}). Figure 12.2 plots the power-on and V_{DD} monitor reset timing. For valid ramp times (less than 1 ms), the power-on reset delay ($T_{PORDelay}$) is typically less than 0.3 ms.

Note: The maximum V_{DD} ramp time is 1 ms; slower ramp times may cause the device to be released from reset before V_{DD} reaches the V_{RST} level.

On exit from a power-on reset, the PORSF flag (RSTSRC.1) is set by hardware to logic 1. When PORSF is set, all of the other reset flags in the RSTSRC Register are indeterminate (PORSF is cleared by all other resets). Since all resets cause program execution to begin at the same location (0x0000), software can read the PORSF flag to determine if a power-up was the cause of reset. The contents of internal data memory should be assumed to be undefined after a power-on reset. The V_{DD} monitor is enabled following a power-on reset.

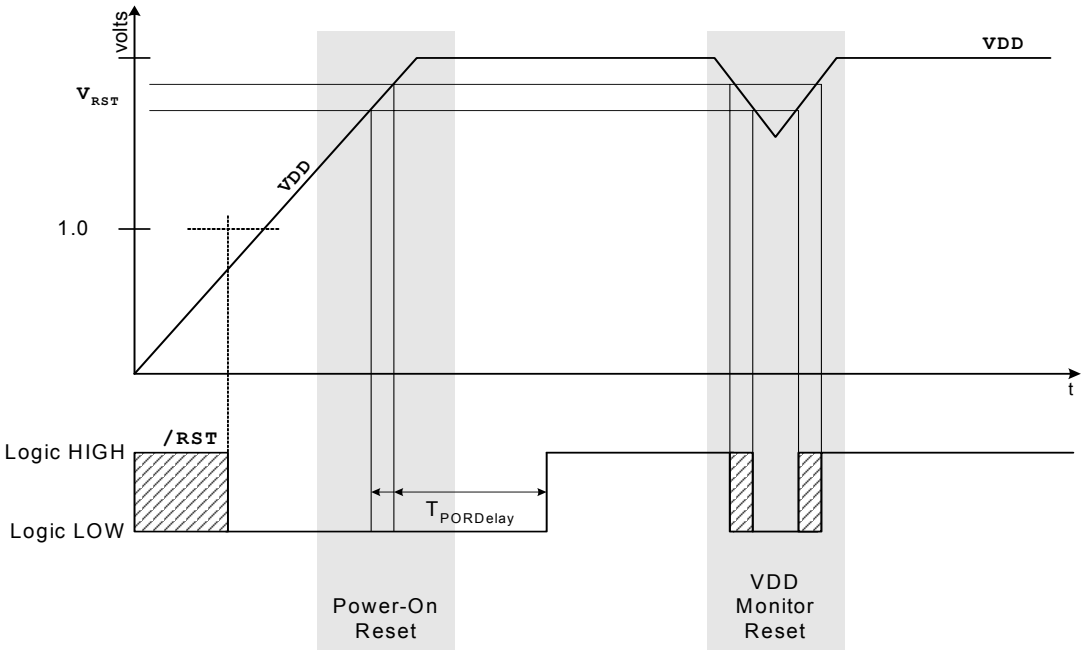


Figure 12.2. Power-On and V_{DD} Monitor Reset Timing

12.2. Power-Fail Reset / V_{DD} Monitor

When a power-down transition or power irregularity causes V_{DD} to drop below V_{RST}, the power supply monitor will drive the $\overline{RST}$ pin low and hold the CIP-51 in a reset state (see Figure 12.2). When V_{DD} returns to a level above V_{RST}, the CIP-51 will be released from the reset state. Note that even though internal data memory contents are not altered by the power-fail reset, it is impossible to determine if V_{DD} dropped below the level required for data retention. If the PORSF flag reads '1', the data may no longer be valid. The V_{DD} monitor is enabled and is not selected as a reset source after power-on resets; however its defined state (enabled/disabled) is not altered by any other reset source. For example, if the V_{DD} monitor is disabled by software, and a software reset is performed, the V_{DD} monitor will still be disabled after the reset. **To protect the integrity of Flash contents, the V_{DD} monitor must be enabled to the higher setting (VDMLVL = '1') and selected as a reset source if software contains routines which erase or write Flash memory. If the V_{DD} monitor is not enabled, any erase or write performed on Flash memory will cause a Flash Error device reset.**

The V_{DD} monitor must be enabled before it is selected as a reset source. Selecting the V_{DD} monitor as a reset source before it is enabled and stabilized may cause a system reset. The procedure for re-enabling the V_{DD} monitor and configuring the V_{DD} monitor as a reset source is shown below:

Step 1. Enable the V_{DD} monitor (VDMEN bit in VDM0CN = '1').
Step 2. Wait for the V_{DD} monitor to stabilize (see Table 12.1 for the V_{DD} Monitor turn-on time).
Note: This delay should be omitted if software contains routines which erase or write Flash memory.
Step 3. Select the V_{DD} monitor as a reset source (PORSF bit in RSTSRC = '1').

See Figure 12.2 for V_{DD} monitor timing; note that the reset delay is not incurred after a V_{DD} monitor reset. See Table 12.1 for complete electrical characteristics of the V_{DD} monitor.

Note: Software should take care not to inadvertently disable the V_{DD} Monitor as a reset source when writing to RSTSRC to enable other reset sources or to trigger a software reset. All writes to RSTSRC should explicitly set PORSF to '1' to keep the V_{DD} Monitor enabled as a reset source.

SFR Definition 12.1. VDDMON: V_{DD} Monitor Control

R/W	R	R/W	R	R	R	R	R	Reset Value
VDDMON	VDDSTAT	VDMLVL	Reserved	Reserved	Reserved	Reserved	Reserved	1v000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xFF

Bit7: **VDDMON**: V_{DD} Monitor Enable.
This bit turns the V_{DD} monitor circuit on/off. The V_{DD} Monitor cannot generate system resets until it is also selected as a reset source in register RSTSRC (SFR Definition 12.2). The V_{DD} Monitor can be allowed to stabilize before it is selected as a reset source. **Selecting the V_{DD} monitor as a reset source before it has stabilized may generate a system reset.** See Table 12.1 for the minimum V_{DD} Monitor turn-on time.
0: V_{DD} Monitor Disabled.
1: V_{DD} Monitor Enabled (default).

Bit6: **VDDSTAT**: V_{DD} Status.
This bit indicates the current power supply status (V_{DD} Monitor output).
0: V_{DD} is at or below the V_{DD} Monitor Threshold.
1: V_{DD} is above the V_{DD} Monitor Threshold.

Bit5: **VDMLVL**: V_{DD} Level Select.
0: V_{DD} Monitor Threshold is set to $V_{RST\text{-}LOW}$ (default).
1: V_{DD} Monitor Threshold is set to $V_{RST\text{-}HIGH}$. This setting is **required** for any system that includes code that writes to and/or erases Flash.

Bits4–0: **RESERVED**. Read = Variable. Write = don't care.

12.3. External Reset

The external $\overline{\text{RST}}$ pin provides a means for external circuitry to force the device into a reset state. Asserting an active-low signal on the $\overline{\text{RST}}$ pin generates a reset; an external pullup and/or decoupling of the $\overline{\text{RST}}$ pin may be necessary to avoid erroneous noise-induced resets. See Table 12.1 for complete $\overline{\text{RST}}$ pin specifications. The PINRSF flag (RSTSRC.0) is set on exit from an external reset.

12.4. Missing Clock Detector Reset

The Missing Clock Detector (MCD) is a one-shot circuit that is triggered by the system clock. If the system clock remains high or low for more than 100 µs, the one-shot will time out and generate a reset. After a MCD reset, the MCDRSF flag (RSTSRC.2) will read '1', signifying the MCD as the reset source; otherwise, this bit reads '0'. Writing a '1' to the MCDRSF bit enables the Missing Clock Detector; writing a '0' disables it. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

12.5. Comparator Reset

Comparator0 can be configured as a reset source by writing a '1' to the C0RSEF flag (RSTSRC.5). Comparator0 should be enabled and allowed to settle prior to writing to C0RSEF to prevent any turn-on chatter on the output from generating an unwanted reset. The Comparator0 reset is active-low: if the non-

inverting input voltage (on CP0+) is less than the inverting input voltage (on CP0-), the device is put into the reset state. After a Comparator0 reset, the C0RSEF flag (RSTSRC.5) will read ‘1’ signifying Comparator0 as the reset source; otherwise, this bit reads ‘0’. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

12.6. PCA Watchdog Timer Reset

The programmable Watchdog Timer (WDT) function of the Programmable Counter Array (PCA) can be used to prevent software from running out of control during a system malfunction. The PCA WDT function can be enabled or disabled by software as described in **Section “20.3. Watchdog Timer Mode” on page 201**; the WDT is enabled and clocked by SYSCLK / 12 following any reset. If a system malfunction prevents user software from updating the WDT, a reset is generated and the WDTRSF bit (RSTSRC.5) is set to ‘1’. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

12.7. Flash Error Reset

If a Flash read/write/erase or program read targets an illegal address, a system reset is generated. This may occur due to any of the following:

- A Flash write or erase is attempted above user code space. This occurs when PSWE is set to ‘1’ and a MOVX write operation targets an address above the Lock Byte address.
- A Flash read is attempted above user code space. This occurs when a MOVC operation targets an address above the Lock Byte address.
- A Program read is attempted above user code space. This occurs when user code attempts to branch to an address above the Lock Byte address.
- A Flash read, write or erase attempt is restricted due to a Flash security setting (see **Section “13.4. Security Options” on page 110**).
- A Flash write or erase is attempted while the V_{DD} Monitor is disabled.

The FERROR bit (RSTSRC.6) is set following a Flash error reset. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

12.8. Software Reset

Software may force a reset by writing a ‘1’ to the SWRSF bit (RSTSRC.4). The SWRSF bit will read ‘1’ following a software forced reset. The state of the $\overline{\text{RST}}$ pin is unaffected by this reset.

SFR Definition 12.2. RSTSRC: Reset Source

R/W	R	R/W	R/W	R	R/W	R/W	R	Reset Value
—	FERROR	C0RSEF	SWRSF	WDTRSF	MCDRSF	PORSF	PINRSF	Variable
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xEF

Note: Software should avoid read modify write instructions when writing values to RSTSRC.

Bit7: **UNUSED**. Read = 1, Write = don't care.
Bit6: **FERROR**: Flash Error Indicator.
0: Source of last reset was not a Flash read/write/erase error.
1: Source of last reset was a Flash read/write/erase error.
Bit5: **C0RSEF**: Comparator0 Reset Enable and Flag.
0: **Read:** Source of last reset was not Comparator0.
Write: Comparator0 is not a reset source.
1: **Read:** Source of last reset was Comparator0.
Write: Comparator0 is a reset source (active-low).
Bit4: **SWRSF**: Software Reset Force and Flag.
0: **Read:** Source of last reset was not a write to the SWRSF bit.
Write: No Effect.
1: **Read:** Source of last was a write to the SWRSF bit.
Write: Forces a system reset.
Bit3: **WDTRSF**: Watchdog Timer Reset Flag.
0: Source of last reset was not a WDT timeout.
1: Source of last reset was a WDT timeout.
Bit2: **MCDRSF**: Missing Clock Detector Flag.
0: **Read:** Source of last reset was not a Missing Clock Detector timeout.
Write: Missing Clock Detector disabled.
1: **Read:** Source of last reset was a Missing Clock Detector timeout.
Write: Missing Clock Detector enabled; triggers a reset if a missing clock condition is detected.
Bit1: **PORSF**: Power-On Reset Force and Flag.
This bit is set anytime a power-on reset occurs. Writing this bit enables/disables the V_{DD} monitor as a reset source. **Note: writing '1' to this bit before the V_{DD} monitor is enabled and stabilized may cause a system reset.** See register VDDMON (SFR Definition 12.1)
0: **Read:** Last reset was not a power-on or V_{DD} monitor reset.
Write: V_{DD} monitor is not a reset source.
1: **Read:** Last reset was a power-on or V_{DD} monitor reset; all other reset flags indeterminate.
Write: V_{DD} monitor is a reset source.
Bit0: **PINRSF**: HW Pin Reset Flag.
0: Source of last reset was not $\overline{\text{RST}}$ pin.
1: Source of last reset was $\overline{\text{RST}}$ pin.

Table 12.1. Reset Electrical Characteristics

–40 to +125 °C unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
$\overline{RST}$ Output Low Voltage	I_{OL} = 8.5 mA, V_{DD} = 2.0 V	—	—	TBD	V
$\overline{RST}$ Input High Voltage		0.7 x V_{DD}	—	—	V
$\overline{RST}$ Input Low Voltage		—	—	0.3 x V_{DD}	V
$\overline{RST}$ Input Pullup Current	$\overline{RST}$ = 0.0 V	—	TBD	TBD	µA
V_{DD} Monitor Threshold ($V_{RST-LOW}$)		TBD	1.7	TBD	V
V_{DD} Monitor Threshold ($V_{RST-HIGH}$)		TBD	2.2	TBD	V
Missing Clock Detector Timeout	Time from last system clock rising edge to reset initiation	TBD	TBD	TBD	µs
Reset Time Delay	Delay between release of any reset source and code execution at location 0x0000	TBD	—	—	µs
Minimum $\overline{RST}$ Low Time to Generate a System Reset		TBD	—	—	µs
V_{DD} Monitor Turn-on Time		TBD	—	—	µs
V_{DD} Monitor Supply Current		—	TBD	TBD	µA

NOTES:

13. Flash Memory

On-chip, re-programmable Flash memory is included for program code and non-volatile data storage. The Flash memory can be programmed in-system through the C2 interface or by software using the MOVX write instruction. Once cleared to logic 0, a Flash bit must be erased to set it back to logic 1. Flash bytes would typically be erased (set to 0xFF) before being reprogrammed. The write and erase operations are automatically timed by hardware for proper execution; data polling to determine the end of the write/erase operations is not required. Code execution is stalled during Flash write/erase operations. Refer to Table 13.2 for complete Flash memory electrical characteristics.

13.1. Programming The Flash Memory

The simplest means of programming the Flash memory is through the C2 interface using programming tools provided by Silicon Laboratories or a third party vendor. This is the only means for programming a non-initialized device. For details on the C2 commands to program Flash memory, see **Section "22. C2 Interface" on page 211**.

To protect the integrity of Flash contents, the V_{DD} monitor must be enabled to the higher setting (VDMLVL = '1') and selected as a reset source if software contains routines which erase or write Flash memory. If the V_{DD} monitor is not enabled, any erase or write performed on Flash memory will cause a Flash Error device reset.

The V_{DD} monitor must be enabled before it is selected as a reset source. Selecting the V_{DD} monitor as a reset source before it is enabled and stabilized may cause a system reset. The procedure for re-enabling the V_{DD} monitor and configuring the V_{DD} monitor as a reset source is shown below:

Step 1. Enable the V_{DD} monitor (VDMEN bit in VDM0CN = '1').
Step 2. Wait for the V_{DD} monitor to stabilize (see Table 12.1 for the V_{DD} Monitor turn-on time).
Note: This delay should be omitted if software contains routines which erase or write Flash memory.
Step 3. Select the V_{DD} monitor as a reset source (PORSF bit in RSTSRC = '1').

13.1.1. Flash Lock and Key Functions

Flash writes and erases by user software are protected with a lock and key function. The Flash Lock and Key Register (FLKEY) must be written with the correct key codes, in sequence, before Flash operations may be performed. The key codes are: 0xA5, 0xF1. The timing does not matter, but the codes must be written in order. If the key codes are written out of order, or the wrong codes are written, Flash writes and erases will be disabled until the next system reset. Flash writes and erases will also be disabled if a Flash write or erase is attempted before the key codes have been written properly. The Flash lock resets after each write or erase; the key codes must be written again before a following Flash operation can be performed. The FLKEY register is detailed in SFR Definition 13.2.

13.1.2. Flash Erase Procedure

The Flash memory can be programmed by software using the MOVX write instruction with the address and data byte to be programmed provided as normal operands. Before writing to Flash memory using MOVX, Flash write operations must be enabled by: (1) setting the PSWE Program Store Write Enable bit (PSCTL.0) to logic 1 (this directs the MOVX writes to target Flash memory); and (2) Writing the Flash key codes in sequence to the Flash Lock register (FLKEY). The PSWE bit remains set until cleared by software.

A write to Flash memory can clear bits to logic 0 but cannot set them; only an erase operation can set bits to logic 1 in Flash. **A byte location to be programmed should be erased before a new value is written.** The Flash memory is organized in 512-byte pages. The erase operation applies to an entire page (setting all bytes in the page to 0xFF). To erase an entire 512-byte page, perform the following steps:

Step 1. Disable interrupts (recommended).
Step 2. Write the first key code to FLKEY: 0xA5.
Step 3. Write the second key code to FLKEY: 0xF1.
Step 4. Set the PSEE bit (register PSCTL).
Step 5. Set the PSWE bit (register PSCTL).
Step 6. Using the MOVX instruction, write a data byte to any location within the 512-byte page to be erased.
Step 7. Clear the PSWE and PSEE bits.
Step 8. Re-enable interrupts.

13.1.3. Flash Write Procedure

Bytes in Flash memory can be written one byte at a time. The FLBWE bit in register PFE0CN (SFR Definition 13.1) controls whether a single byte or a block of two bytes is written to Flash during a write operation. When FLBWE is cleared to ‘0’, the Flash will be written one byte at a time. When FLBWE is set to ‘1’, the Flash will be written in two-byte blocks. Block writes are performed in the same amount of time as single-byte writes, which can save time when storing large amounts of data to Flash memory.

During a single-byte write to Flash, bytes are written individually, and a Flash write will be performed after each MOVX write instruction. The recommended procedure for writing Flash in single bytes is:

Step 1. Disable interrupts.
Step 2. Clear the FLBWE bit (register PFE0CN) to select single-byte write mode.
Step 3. Write the first key code to FLKEY: 0xA5.
Step 4. Write the second key code to FLKEY: 0xF1.
Step 5. Set the PSWE bit (register PSCTL).
Step 6. Clear the PSEE bit (register PSCTL).
Step 7. Using the MOVX instruction, write a single data byte to the desired location within the 512-byte sector.
Step 8. Clear the PSWE bit.
Step 9. Re-enable interrupts.

Steps 3–8 must be repeated for each byte to be written.

For block Flash writes, the Flash write procedure is only performed after the last byte of each block is written with the MOVX write instruction. A Flash write block is two bytes long, from even addresses to odd addresses. Writes must be performed sequentially (i.e. addresses ending in 0b and 1b must be written in order). The Flash write will be performed following the MOVX write that targets the address ending in 1b. If a byte in the block does not need to be updated in Flash, it should be written to 0xFF. The recommended procedure for writing Flash in blocks is:

Step 1. Disable interrupts.
Step 2. Set the FLBWE bit (register PFE0CN) to select block write mode.
Step 3. Write the first key code to FLKEY: 0xA5.
Step 4. Write the second key code to FLKEY: 0xF1.
Step 5. Set the PSWE bit (register PSCTL).
Step 6. Clear the PSEE bit (register PSCTL).
Step 7. Using the MOVX instruction, write the first data byte to the even block location (ending in 0b).
Step 8. Clear the PSWE bit (register PSCTL).
Step 9. Write the first key code to FLKEY: 0xA5.
Step 10. Write the second key code to FLKEY: 0xF1.
Step 11. Set the PSWE bit (register PSCTL).
Step 12. Clear the PSEE bit (register PSCTL).
Step 13. Using the MOVX instruction, write the second data byte to the odd block location (ending in 1b).
Step 14. Clear the PSWE bit (register PSCTL).
Step 15. Re-enable interrupts.

Steps 3–14 must be repeated for each block to be written.

13.2. Flash Write and Erase Guidelines

Any system which contains routines which write or erase Flash memory from software involves some risk that the write or erase routines will execute unintentionally if the CPU is operating outside its specified operating range of V_{DD}, system clock frequency, or temperature. This accidental execution of Flash modifying code can result in alteration of Flash memory contents causing a system failure that is only recoverable by re-Flashing the code in the device.

The following guidelines are recommended for any system which contains routines which write or erase Flash from code.

13.2.1. V_{DD} Maintenance and the V_{DD} monitor

1. If the system power supply is subject to voltage or current "spikes," add sufficient transient protection devices to the power supply to ensure that the supply voltages listed in the Absolute Maximum Ratings table are not exceeded.
2. Make certain that the minimum V_{DD} rise time specification of 1 ms is met. If the system cannot meet this rise time specification, then add an external V_{DD} brownout circuit to the $\overline{RST}$ pin of the device that holds the device in reset until V_{DD} reaches 2.7 V and re-asserts $\overline{RST}$ if V_{DD} drops below 2.7 V.
3. Enable the on-chip V_{DD} monitor and enable the V_{DD} monitor as a reset source as early in code as possible. This should be the first set of instructions executed after the Reset Vector. For 'C'-based systems, this will involve modifying the startup code added by the 'C' compiler. See your compiler documentation for more details. Make certain that there are no delays in software between enabling the V_{DD} monitor and enabling the V_{DD} monitor as a reset source. Code examples showing this can be found in “AN201: Writing to Flash from Firmware", available from the Silicon Laboratories web site.
4. As an added precaution, explicitly enable the V_{DD} monitor and enable the V_{DD} monitor as a reset source inside the functions that write and erase Flash memory. The V_{DD} monitor enable instructions should be placed just after the instruction to set PSWE to a '1', but before the Flash write or erase operation instruction.
5. Make certain that all writes to the RSTSRC (Reset Sources) register use direct assignment operators and explicitly DO NOT use the bit-wise operators (such as AND or OR). For example, "RSTSRC = 0x02" is correct. "RSTSRC |= 0x02" is incorrect.
6. Make certain that all writes to the RSTSRC register explicitly set the PORSF bit to a '1'. Areas to check are initialization code which enables other reset sources, such as the Missing Clock Detector or Comparator, for example, and instructions which force a Software Reset. A global search on "RSTSRC" can quickly verify this.

13.2.2. PSWE Maintenance

7. Reduce the number of places in code where the PSWE bit (b0 in PSCTL) is set to a '1'. There should be exactly one routine in code that sets PSWE to a '1' to write Flash bytes and one routine in code that sets PSWE and PSEE both to a '1' to erase Flash pages.
8. Minimize the number of variable accesses while PSWE is set to a '1'. Handle pointer address updates and loop variable maintenance outside the "PSWE = 1; ... PSWE = 0;" area. Code examples showing this can be found in *AN201, "Writing to Flash from Firmware"*, available from the Silicon Laboratories web site.
9. Disable interrupts prior to setting PSWE to a '1' and leave them disabled until after PSWE has been reset to '0'. Any interrupts posted during the Flash write or erase operation will be serviced in priority order after the Flash operation has been completed and interrupts have been re-enabled by software.
10. Make certain that the Flash write and erase pointer variables are not located in XRAM. See your compiler documentation for instructions regarding how to explicitly locate variables in different memory areas.
11. Add address bounds checking to the routines that write or erase Flash memory to ensure that a routine called with an illegal address does not result in modification of the Flash.

13.2.3. System Clock

12. If operating from an external crystal, be advised that crystal performance is susceptible to electrical interference and is sensitive to layout and to changes in temperature. If the system is operating in an electrically noisy environment, use the internal oscillator or use an external CMOS clock.
13. If operating from the external oscillator, switch to the internal oscillator during Flash write or erase operations. The external oscillator can continue to run, and the CPU can switch back to the external oscillator after the Flash operation has completed.

Additional Flash recommendations and example code can be found in *AN201, "Writing to Flash from Firmware"*, available from the Silicon Laboratories web site.

13.3. Non-volatile Data Storage

The Flash memory can be used for non-volatile data storage as well as program code. This allows data such as calibration coefficients to be calculated and stored at run time. Data is written using the MOVX write instruction and read using the MOVC instruction. Note: MOVX read instructions always target XRAM.

13.4. Security Options

The CIP-51 provides security options to protect the Flash memory from inadvertent modification by software as well as to prevent the viewing of proprietary program code and constants. The Program Store Write Enable (bit PSWE in register PSCTL) and the Program Store Erase Enable (bit PSEE in register PSCTL) bits protect the Flash memory from accidental modification by software. PSWE must be explicitly set to '1' before software can modify the Flash memory; both PSWE and PSEE must be set to '1' before software can erase Flash memory. Additional security features prevent proprietary program code and data constants from being read or altered across the C2 interface.

A Security Lock Byte located at the last byte of Flash user space offers protection of the Flash program memory from access (reads, writes, or erases) by unprotected code or the C2 interface. The Flash security mechanism allows the user to lock *n* 512-byte Flash pages, starting at page 0 (addresses 0x0000 to 0x01FF), where *n* is the 1's complement number represented by the Security Lock Byte. **Note that the page containing the Flash Security Lock Byte is unlocked when no other Flash pages are locked (all bits of the Lock Byte are '1') and locked when any other Flash pages are locked (any bit of the Lock Byte is '0').** See example below.

Security Lock Byte:	11111101b
1's Complement:	00000010b
Flash pages locked:	3 (First two Flash pages + Lock Byte Page)
Addresses locked:	0x0000 to 0x03FF (first two Flash pages) and 0x7C00 to 0x7DFF (Lock Byte Page)

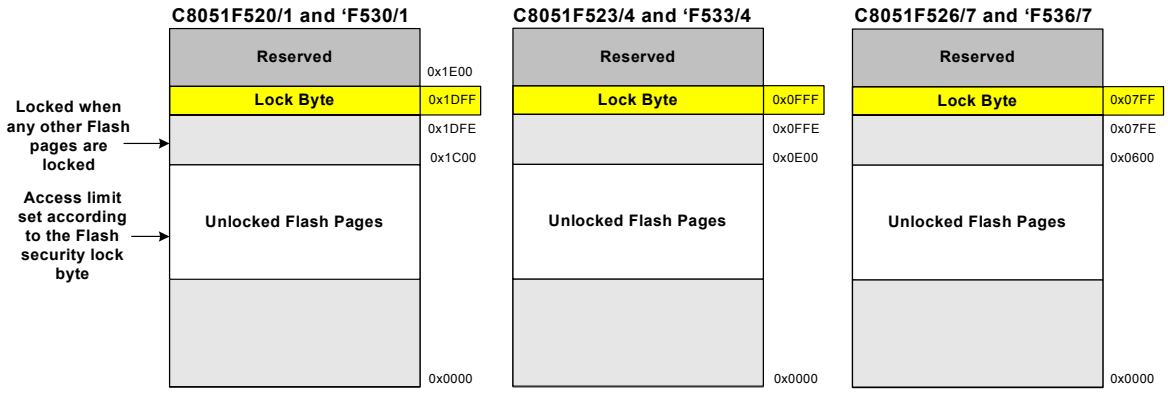


Figure 13.1. Flash Program Memory Map

The level of Flash security depends on the Flash access method. The three Flash access methods that can be restricted are reads, writes, and erases from the C2 debug interface, user firmware executing on unlocked pages, and user firmware executing on locked pages.

Accessing Flash **from the C2 debug interface**:

1. Any unlocked page may be read, written, or erased.
2. Locked pages cannot be read, written, or erased.
3. The page containing the Lock Byte may be read, written, or erased if it is unlocked.
4. Reading the contents of the Lock Byte is always permitted only if no pages are locked.
5. Locking additional pages (changing ‘1’s to ‘0’s in the Lock Byte) is not permitted.
6. Unlocking Flash pages (changing ‘0’s to ‘1’s in the Lock Byte) requires the C2 Device Erase command, which erases all Flash pages including the page containing the Lock Byte and the Lock Byte itself.
7. The Reserved Area cannot be read, written, or erased.

Accessing Flash **from user firmware** executing from an **unlocked page**:

1. Any unlocked page except the page containing the Lock Byte may be read, written, or erased.
2. Locked pages cannot be read, written, or erased. An erase attempt on the page containing the Lock Byte will result in a Flash Error device reset.
3. **The page containing the Lock Byte cannot be erased.** It may be read or written only if it is unlocked. An erase attempt on the page containing the Lock Byte will result in a Flash Error device reset.
4. Reading the contents of the Lock Byte is always permitted.
5. Locking additional pages (changing ‘1’s to ‘0’s in the Lock Byte) is not permitted.
6. Unlocking Flash pages (changing ‘0’s to ‘1’s in the Lock Byte) is not permitted.
7. The Reserved Area cannot be read, written, or erased. Any attempt to access the reserved area, or any other locked page, will result in a Flash Error device reset.

Accessing Flash **from user firmware** executing from a **locked page**:

1. Any unlocked page except the page containing the Lock Byte may be read, written, or erased.
2. Any locked page except the page containing the Lock Byte may be read, written, or erased. An erase attempt on the page containing the Lock Byte will result in a Flash Error device reset.
3. **The page containing the Lock Byte cannot be erased.** It may only be read or written. An erase attempt on the page containing the Lock Byte will result in a Flash Error device reset.
4. Reading the contents of the Lock Byte is always permitted.
5. Locking additional pages (changing ‘1’s to ‘0’s in the Lock Byte) is not permitted.
6. Unlocking Flash pages (changing ‘0’s to ‘1’s in the Lock Byte) is not permitted.
7. The Reserved Area cannot be read, written, or erased. Any attempt to access the reserved area, or any other locked page, will result in a Flash Error device reset.

The level of Flash security depends on the Flash access method. The three Flash access methods that can be restricted are reads, writes, and erases from the C2 debug interface, user firmware executing on unlocked pages, and user firmware executing on locked pages. Table 13.1 summarizes the Flash security features of the 'F330/1/2/3/4/5 devices.

Table 13.1. Flash Security Summary

Action	C2 Debug Interface	User Firmware executing from:	
		an unlocked page	a locked page
Read, Write or Erase unlocked pages (except page with Lock Byte)	Permitted	Permitted	Permitted
Read, Write or Erase locked pages (except page with Lock Byte)	Not Permitted	Flash Error Reset	Permitted
Read or Write page containing Lock Byte (if no pages are locked)	Permitted	Permitted	Permitted
Read or Write page containing Lock Byte (if any page is locked)	Not Permitted	Flash Error Reset	Permitted
Read contents of Lock Byte (if no pages are locked)	Permitted	Permitted	Permitted
Read contents of Lock Byte (if any page is locked)	Not Permitted	Flash Error Reset	Permitted
Erase page containing Lock Byte (if no pages are locked)	Permitted	Flash Error Reset	Flash Error Reset
Erase page containing Lock Byte - Unlock all pages (if any page is locked)	C2 Device Erase Only	Flash Error Reset	Flash Error Reset
Lock additional pages (change '1's to '0's in the Lock Byte)	Not Permitted	Flash Error Reset	Flash Error Reset
Unlock individual pages (change '0's to '1's in the Lock Byte)	Not Permitted	Flash Error Reset	Flash Error Reset
Read, Write or Erase Reserved Area	Not Permitted	Flash Error Reset	Flash Error Reset

C2 Device Erase - Erases all Flash pages including the page containing the Lock Byte.

Flash Error Reset - Not permitted; Causes Flash Error Device Reset (FERROR bit in RSTSRC is '1' after reset).

- All prohibited operations that are performed via the C2 interface are ignored (do not cause device reset).
- Locking any Flash page also locks the page containing the Lock Byte.
- Once written to, the Lock Byte cannot be modified except by performing a C2 Device Erase.
- If user code writes to the Lock Byte, the Lock does not take effect until the next device reset.

SFR Definition 13.1. PSCTL: Program Store R/W Control

R	R	R	R	R	R	R/W	R/W	Reset Value
-	-	-	-	-	-	PSEE	PSWE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x8F

Bits7–2: **UNUSED**: Read = 000000b, Write = don't care.

Bit1: **PSEE**: Program Store Erase Enable
Setting this bit (in combination with PSWE) allows an entire page of Flash program memory to be erased. If this bit is logic 1 and Flash writes are enabled (PSWE is logic 1), a write to Flash memory using the MOVX instruction will erase the entire page that contains the location addressed by the MOVX instruction. The value of the data byte written does not matter.
0: Flash program memory erasure disabled.
1: Flash program memory erasure enabled.

Bit0: **PSWE**: Program Store Write Enable
Setting this bit allows writing a byte of data to the Flash program memory using the MOVX write instruction. The Flash location should be erased before writing data.
0: Writes to Flash program memory disabled.
1: Writes to Flash program memory enabled; the MOVX write instruction targets Flash memory.

SFR Definition 13.2. FLKEY: Flash Lock and Key

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xB7

Bits7–0: **FLKEY**: Flash Lock and Key Register
Write:
This register provides a lock and key function for Flash erasures and writes. Flash writes and erases are enabled by writing 0xA5 followed by 0xF1 to the FLKEY register. Flash writes and erases are automatically disabled after the next write or erase is complete. If any writes to FLKEY are performed incorrectly, or if a Flash write or erase operation is attempted while these operations are disabled, the Flash will be permanently locked from writes or erasures until the next device reset. If an application never writes to Flash, it can intentionally lock the Flash by writing a non-0xA5 value to FLKEY from software.
Read:
When read, bits 1–0 indicate the current Flash lock state.
00: Flash is write/erase locked.
01: The first key code has been written (0xA5).
10: Flash is unlocked (writes/erases allowed).
11: Flash writes/erases disabled until the next reset.

Table 13.2. Flash Electrical Characteristics

V_{DD} = 1.8 to 2.75 V; –40 to +125 ºC unless otherwise specified

Parameter	Conditions	Min	Typ	Max	Units
Flash Size	C8051F520/1 and ‘F530/1	7680	—	—	bytes
	C8051F523/4 and ‘F533/4	4096			
	C8051F526/7 and ‘F536/7	2048			
Endurance	V_{DD} is 2.2 V or greater	40 k	150 k	—	Erase/Write
Erase Cycle Time		32	40	48	ms
Write Cycle Time		76	92	114	µs
V_{DD}	Write/Erase Operations	2.25	—	—	V

14. Port Input/Output

Digital and analog resources are available through up to 16 I/O pins. Port pins are organized as two or one byte-wide Ports. Each of the Port pins can be defined as general-purpose I/O (GPIO) or analog input/output; Port pins P0.0 - P1.7 can be assigned to one of the internal digital resources as shown in Figure 14.3. The designer has complete control over which functions are assigned, limited only by the number of physical I/O pins. This resource assignment flexibility is achieved through the use of a Priority Crossbar Decoder. Note that the state of a Port I/O pin can always be read in the corresponding Port latch, regardless of the Crossbar settings.

The Crossbar assigns the selected internal digital resources to the I/O pins based on the peripheral priority order of the Priority Decoder (Figure 14.3 and Figure 14.4). The registers XBR0 and XBR1, defined in SFR Definition 14.1 and SFR Definition 14.2, are used to select internal digital functions.

Port I/Os on P0 are 5.25 V tolerant over the operating range of V_{IO}. Figure 14.2 shows the Port cell circuit. The Port I/O cells are configured as either push-pull or open-drain in the Port Output Mode registers (PnMDOUT, where n = 0,1). Complete Electrical Specifications for Port I/O are given in Table 14.1 on page 130.

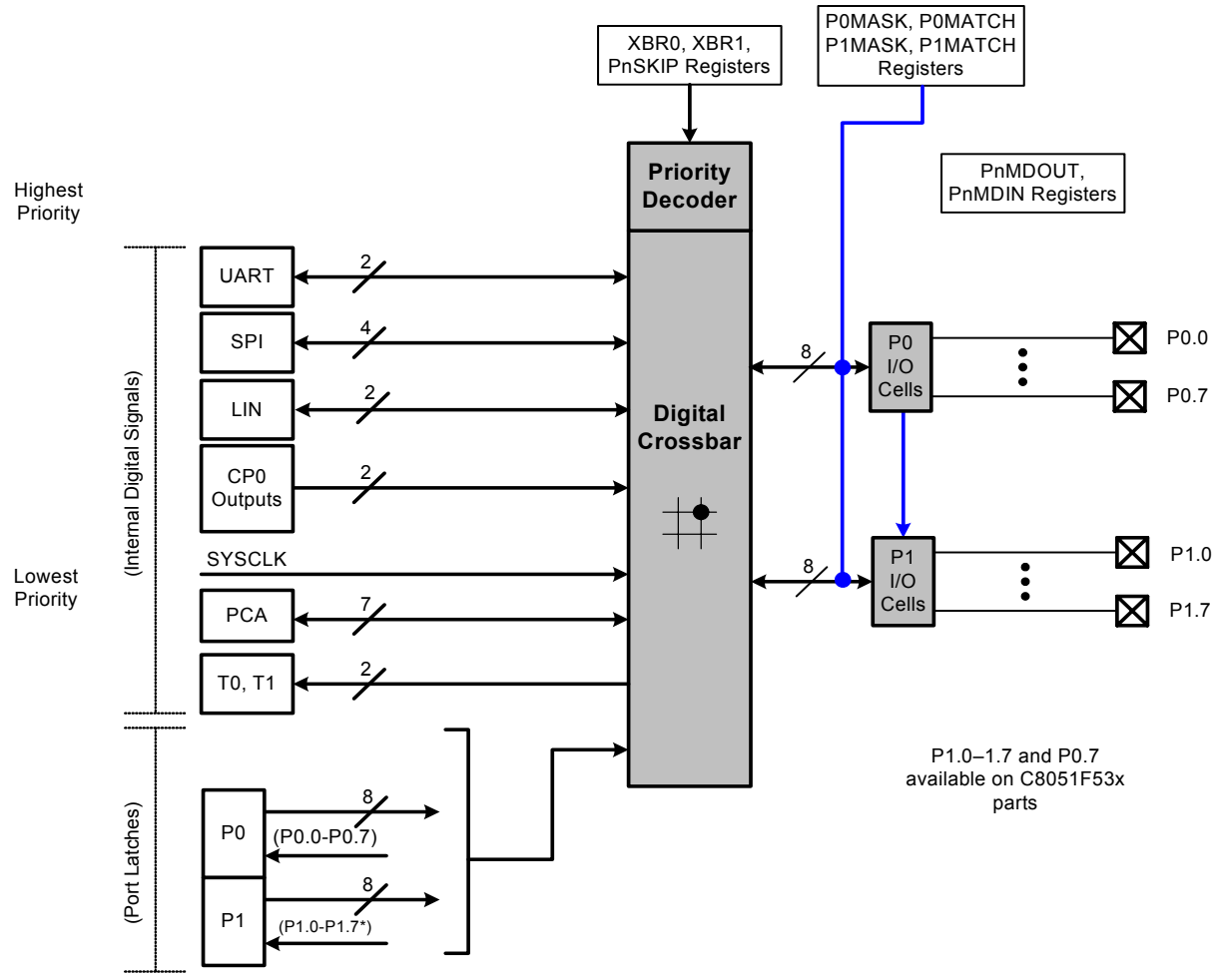


Figure 14.1. Port I/O Functional Block Diagram

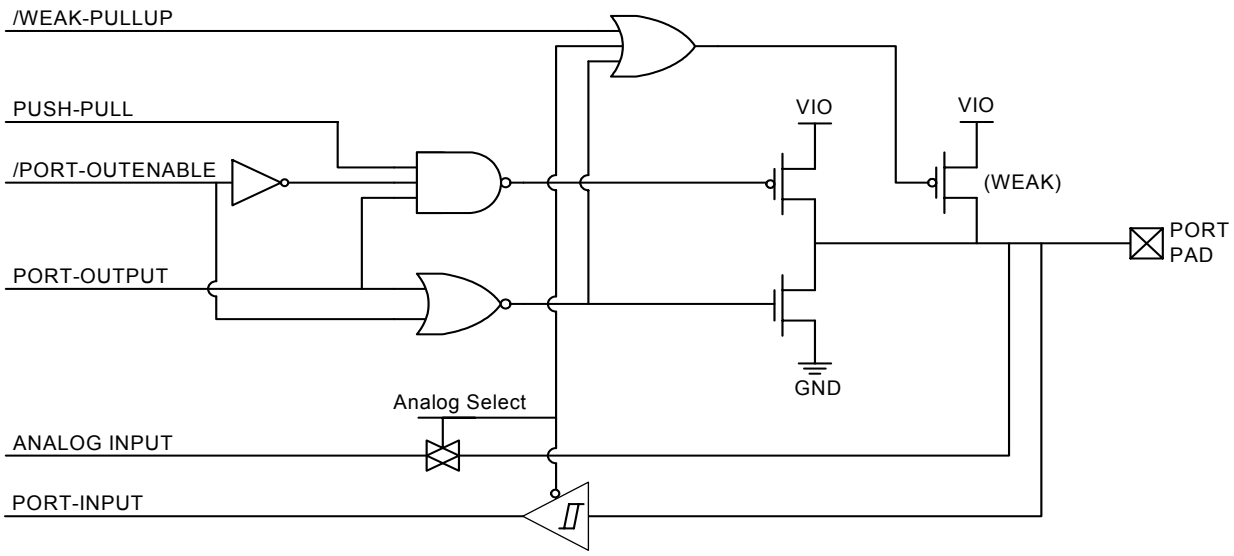
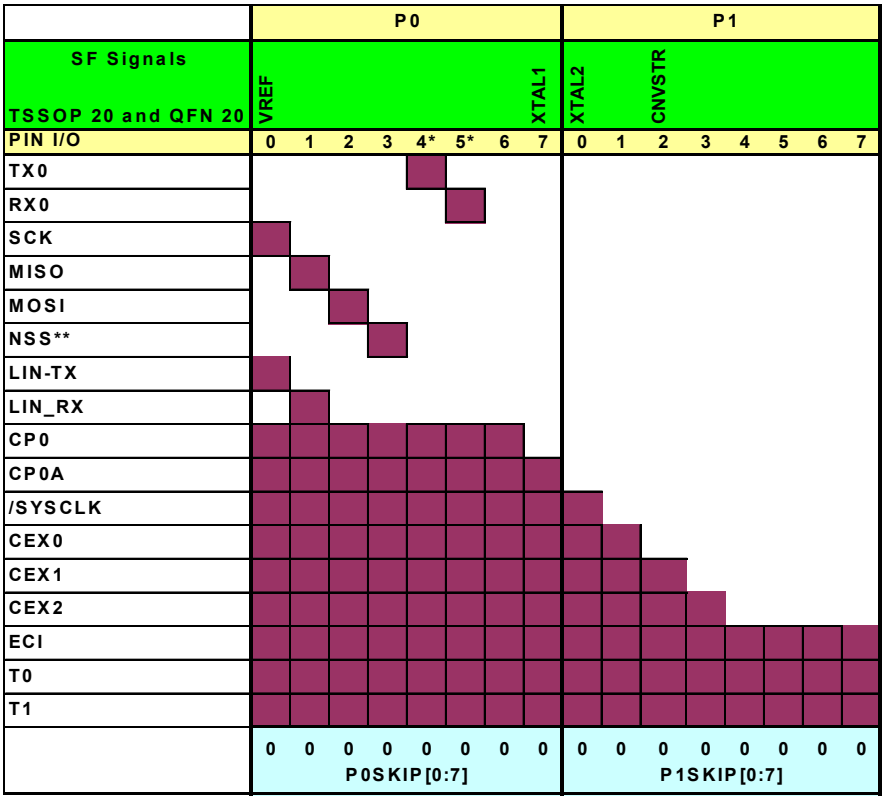


Figure 14.2. Port I/O Cell Block Diagram

14.1. Priority Crossbar Decoder

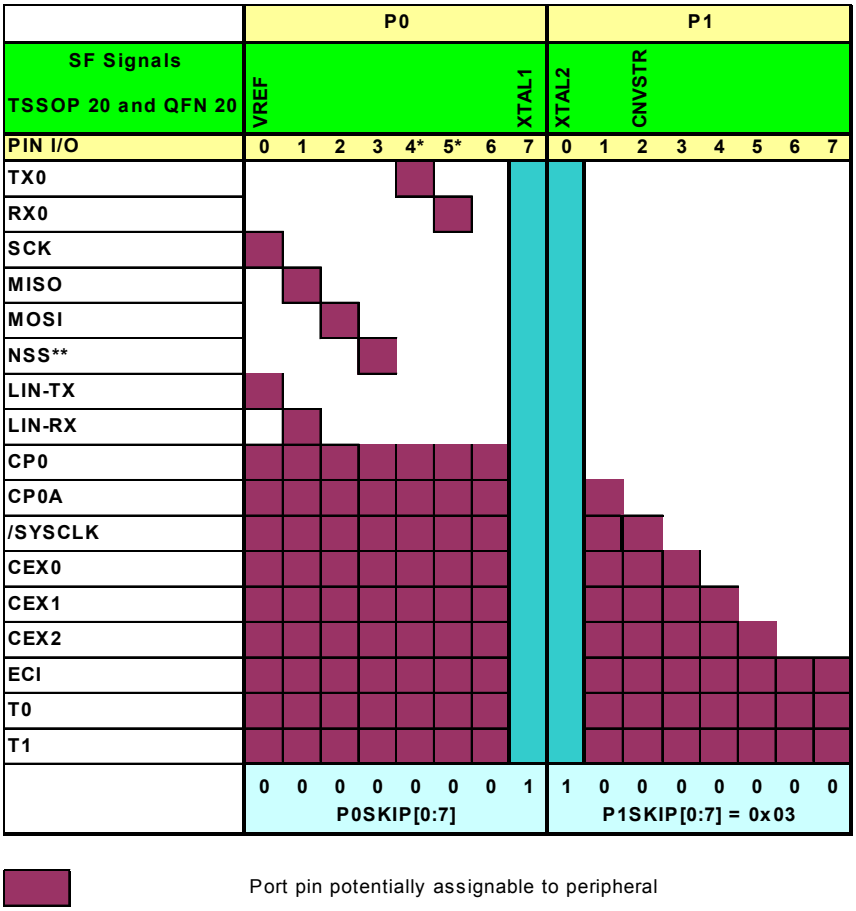
The Priority Crossbar Decoder (Figure 14.3) assigns a priority to each I/O function, starting at the top with UART0. When a digital resource is selected, the least-significant unassigned Port pin is assigned to that resource (excluding UART0, which will be assigned to pins P0.4 and P0.5). If a Port pin is assigned, the Crossbar skips that pin when assigning the next selected resource. Additionally, the Crossbar will skip Port pins whose associated bits in the PnSKIP registers are set. The PnSKIP registers allow software to skip Port pins that are to be used for analog input, dedicated functions, or GPIO.

Important Note on Crossbar Configuration: If a Port pin is claimed by a peripheral without use of the Crossbar, its corresponding PnSKIP bit should be set. This applies to P1.0 and/or P0.7 ('F530) or P0.2 and/or P0.3 ('F5250) for the external oscillator, P0.0 for V_{REF}, P1.2 ('F530) or P0.5 ('F530) for the external CNVSTR signal, and any selected ADC or comparator inputs. The Crossbar skips selected pins as if they were already assigned, and moves to the next unassigned pin. Figure 14.3 shows the Crossbar Decoder priority with no Port pins skipped (P0SKIP, P1SKIP); Figure 14.4 shows the Crossbar Decoder priority with the XTAL1 (P1.0) and XTAL2 (P1.1) pins skipped (P1SKIP = 0x03).



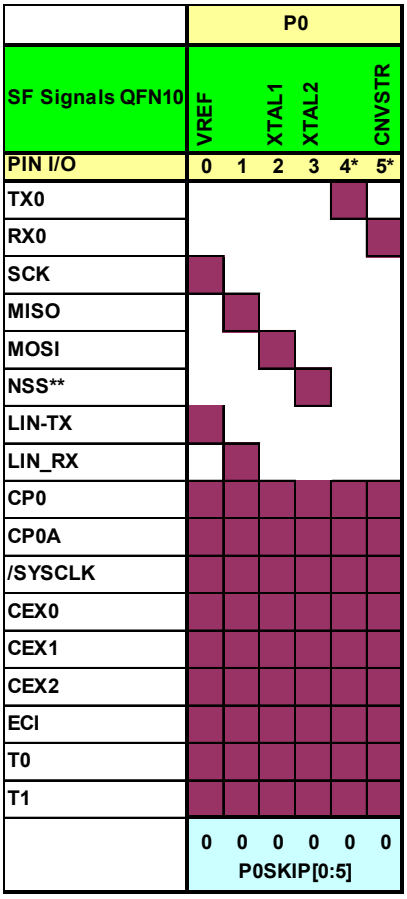
***Note:** Refer to **Section "21. Revision Specific Behavior" on page 209**.
****Note:** 4-Wire SPI Only.

Figure 14.3. Crossbar Priority Decoder with No Pins Skipped (TSSOP 20 and QFN 20)



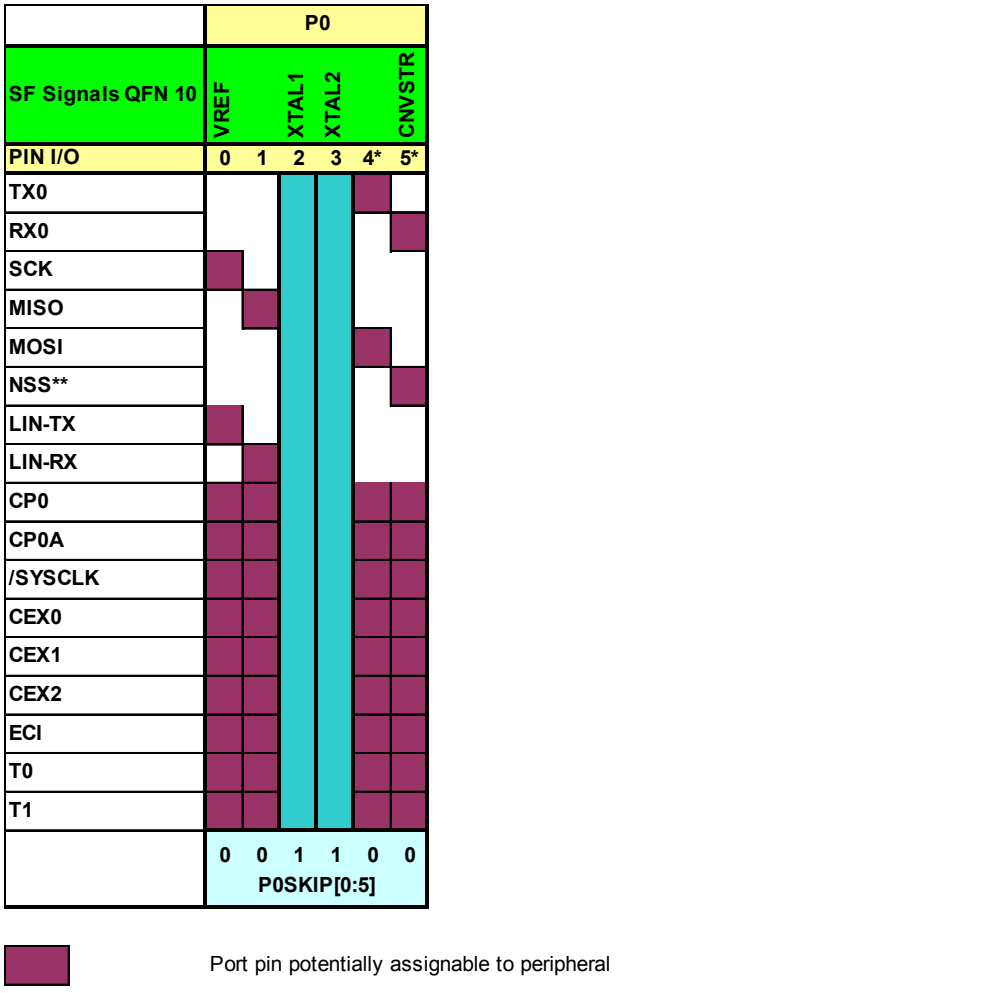
***Note:** Refer to **Section "21. Revision Specific Behavior" on page 209**.
****Note:** 4-Wire SPI Only.

Figure 14.4. Crossbar Priority Decoder with Crystal Pins Skipped (TSSOP 20 and QFN 20)



***Note:** Refer to **Section "21. Revision Specific Behavior" on page 209**.
****Note:** 4-Wire SPI Only.

Figure 14.5. Crossbar Priority Decoder with No Pins Skipped (QFN 10)



***Note:** Refer to **Section "21. Revision Specific Behavior" on page 209**.
****Note:** 4-Wire SPI Only.

Figure 14.6. Crossbar Priority Decoder with Crystal Pins Skipped (QFN 10)

Registers XBR0 and XBR1 are used to assign the digital I/O resources to the physical I/O Port pins. Note that when the SMBus is selected, the Crossbar assigns both pins associated with the SMBus (SDA and SCL); when the UART is selected, the Crossbar assigns both pins associated with the UART (TX and RX). UART0 pin assignments are fixed for bootloading purposes: UART TX0 is always assigned to P0.3 or P0.4*; UART RX0 is always assigned to P0.4 or P0.5*. Standard Port I/Os appear contiguously starting at P0.0 after prioritized functions and skipped pins are assigned.

***Note:** Refer to **Section "21. Revision Specific Behavior" on page 209**.

Important Note: The SPI can be operated in either 3-wire or 4-wire modes, depending on the state of the NSSMD1–NSSMD0 bits in register SPI0CN. According to the SPI mode, the NSS signal may or may not be routed to a Port pin.

14.2. Port I/O Initialization

Port I/O initialization consists of the following steps:

Step 1. Select the input mode (analog or digital) for all Port pins, using the Port Input Mode register (PnMDIN).
Step 2. Select the output mode (open-drain or push-pull) for all Port pins, using the Port Output Mode register (PnMDOUT).
Step 3. Select any pins to be skipped by the I/O Crossbar using the Port Skip registers (PnSKIP).
Step 4. Assign Port pins to desired peripherals using the XBRn registers.
Step 5. Enable the Crossbar (XBARE = ‘1’).

All Port pins must be configured as either analog or digital inputs. Any pins to be used as Comparator or ADC inputs should be configured as an analog inputs. When a pin is configured as an analog input, its weak pullup, digital driver, and digital receiver are disabled. This process saves power and reduces noise on the analog input. Pins configured as digital inputs may still be used by analog peripherals; however, this practice is not recommended.

Additionally, all analog input pins should be configured to be skipped by the Crossbar (accomplished by setting the associated bits in PnSKIP). Port input mode is set in the PnMDIN register, where a ‘1’ indicates a digital input, and a ‘0’ indicates an analog input. All pins default to digital inputs on reset. See SFR Definition 14.4 for the PnMDIN register details.

Important Note: Port 0 and Port 1 pins are 5.25 V tolerant across the operating range of V_{REGIN}.

The output driver characteristics of the I/O pins are defined using the Port Output Mode registers (PnMDOUT). Each Port Output driver can be configured as either open drain or push-pull. This selection is required even for the digital resources selected in the XBRn registers, and is not automatic. When the WEAKPUD bit in XBR1 is ‘0’, a weak pullup is enabled for all Port I/O configured as open-drain. WEAKPUD does not affect the push-pull Port I/O. Furthermore, the weak pullup is turned off on an output that is driving a ‘0’ and for pins configured for analog input mode to avoid unnecessary power dissipation.

Registers XBR0 and XBR1 must be loaded with the appropriate values to select the digital I/O functions required by the design. Setting the XBARE bit in XBR1 to ‘1’ enables the Crossbar. Until the Crossbar is enabled, the external pins remain as standard Port I/O (in input mode), regardless of the XBRn Register settings. For given XBRn Register settings, one can determine the I/O pin-out using the Priority Decode Table.

The Crossbar must be enabled to use Port pins as standard Port I/O in output mode. **Port output drivers are disabled while the Crossbar is disabled.**

SFR Definition 14.1. XBR0: Port I/O Crossbar Register 0

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	CP0AE	CP0E	SYSCKE	LINE	SPI0E	URT0E	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xE1

Bit7–6: **RESERVED**. Read = 0b; Must write 0b.
Bit5: **CP0AE**: Comparator0 Asynchronous Output Enable
0: Asynchronous CP0 unavailable at Port pin.
1: Asynchronous CP0 routed to Port pin.
Bit4: **CP0E**: Comparator0 Output Enable
0: CP0 unavailable at Port pin.
1: CP0 routed to Port pin.
Bit3: **SYSCKE**: /SYSCLK Output Enable
0: /SYSCLK unavailable at Port pin.
1: /SYSCLK output routed to Port pin.
Bit2: **LINE**. Lin Output Enable
Bit1: **SPI0E**: SPI I/O Enable
0: SPI I/O unavailable at Port pins.
1: SPI I/O routed to Port pins. Note that the SPI can be assigned either 3 or 4 GPIO pins.
Bit0: **URT0E**: UART I/O Output Enable
0: UART I/O unavailable at Port pin.
1: UART TX0, RX0 routed to Port pins (P0.3 and P0.4) or (P0.4 and P0.5).

Note: Please refer to Section "**21. Revision Specific Behavior**" on page **209**.

SFR Definition 14.2. XBR1: Port I/O Crossbar Register 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
WEAKPUD	XBARE	T1E	T0E	ECIE	—	PCA0ME		00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xE2

Bit7: **WEAKPUD**: Port I/O Weak Pullup Disable.
0: Weak Pullups enabled (except for Ports whose I/O are configured as analog input).
1: Weak Pullups disabled.
Bit6: **XBARE**: Crossbar Enable.
0: Crossbar disabled.
1: Crossbar enabled.
Bit5: **T1E**: T1 Enable
0: T1 unavailable at Port pin.
1: T1 routed to Port pin.
Bit4: **T0E**: T0 Enable
0: T0 unavailable at Port pin.
1: T0 routed to Port pin.
Bit3: **ECIE**: PCA0 External Counter Input Enable
0: ECI unavailable at Port pin.
1: ECI routed to Port pin.
Bit2: Reserved.
Bits1–0: **PCA0ME**: PCA Module I/O Enable Bits.
00: All PCA I/O unavailable at Port pins.
01: CEX0 routed to Port pin.
10: CEX0, CEX1 routed to Port pins.
11: CEX0, CEX1, CEX2 routed to Port pins.

14.3. General Purpose Port I/O

Port pins that remain unassigned by the Crossbar and are not used by analog peripherals can be used for general purpose I/O. Ports P0**–**P1 are accessed through corresponding special function registers (SFRs) that are both byte addressable and bit addressable. When writing to a Port, the value written to the SFR is latched to maintain the output data value at each pin. When reading, the logic levels of the Port's input pins are returned regardless of the XBRn settings (i.e., even when the pin is assigned to another signal by the Crossbar, the Port register can always read its corresponding Port I/O pin). The exception to this is the execution of the read-modify-write instructions that target a Port Latch register as the destination. The read-modify-write instructions when operating on a Port SFR are the following: ANL, ORL, XRL, JBC, CPL, INC, DEC, DJNZ and MOV, CLR or SETB, when the destination is an individual bit in a Port SFR. For these instructions, the value of the latch register (not the pin) is read, modified, and written back to the SFR.

In addition to performing general purpose I/O, P0 and P1 can generate a port match event if the logic levels of the Port's input pins match a software controlled value. A port match event is generated if

(P0 & P0MASK) does not equal (P0MATCH & P0MASK) or if (P1 & P1MASK) does not equal (P1MATCH & P1MASK). This allows Software to be notified if a certain change or pattern occurs on P0 or P1 input pins regardless of the XBRn settings. A port match event can cause an interrupt if EMAT (EIE2.1) is set to '1' or cause the internal oscillator to awaken from SUSPEND mode. See Section "**15.1.1. Internal Oscillator Suspend Mode**" on page **132** for more information.

SFR Definition 14.3. P0: Port0

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0x80

Bits7–0: **P0.[7:0]**
Write - Output appears on I/O pins per Crossbar Registers.
0: Logic Low Output.
1: Logic High Output (high impedance if corresponding P0MDOUT.n bit = 0).
Read - Always reads '0' if selected as analog input in register P0MDIN. Directly reads Port pin when configured as digital input.
0: P0.n pin is logic low.
1: P0.n pin is logic high.

SFR Definition 14.4. P0MDIN: Port0 Input Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xF1

Bits7–0: Analog Input Configuration Bits for P0.7–P0.0 (respectively).
Port pins configured as analog inputs have their weak pullup, digital driver, and digital receiver disabled.
0: Corresponding P0.n pin is configured as an analog input.
1: Corresponding P0.n pin is not configured as an analog input.

SFR Definition 14.5. P0MDOUT: Port0 Output Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xA4

Bits7–0: Output Configuration Bits for P0.7–P0.0 (respectively): ignored if corresponding bit in register P0MDIN is logic 0.
0: Corresponding P0.n Output is open-drain.
1: Corresponding P0.n Output is push-pull.

(Note: When SDA and SCL appear on any of the Port I/O, each are open-drain regardless of the value of P0MDOUT).

SFR Definition 14.6. P0SKIP: Port0 Skip

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xD4

Bits7–0: **P0SKIP[7:0]**: Port0 Crossbar Skip Enable Bits.
These bits select Port pins to be skipped by the Crossbar Decoder. Port pins used as analog inputs (for ADC or Comparator) or used as special functions (V_{REF} input, external oscillator circuit, CNVSTR input) should be skipped by the Crossbar.
0: Corresponding P0.n pin is not skipped by the Crossbar.
1: Corresponding P0.n pin is skipped by the Crossbar.

SFR Definition 14.7. P0MAT: Port0 Match

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xD7

Bits7–0: **P0MAT[7:0]**: Port0 Match Value.
These bits control the value that unmasked P0 Port pins are compared against. A Port Match event is generated if (P0 & P0MASK) does not equal (P0MAT & P0MASK).

SFR Definition 14.8. P0MASK: Port0 Mask

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xC7

Bits7–0: **P0MASK[7:0]**: Port0 Mask Value.
These bits select which Port pins will be compared to the value stored in P0MAT.
0: Corresponding P0.n pin is ignored and cannot cause a Port Match event.
1: Corresponding P0.n pin is compared to the corresponding bit in P0MAT.

SFR Definition 14.9. P1: Port1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0x90

Bits7–0: **P1.[7:0]**
Write - Output appears on I/O pins per Crossbar Registers.
0: Logic Low Output.
1: Logic High Output (high impedance if corresponding P1MDOUT.n bit = 0).
Read - Always reads '0' if selected as analog input in register P1MDIN. Directly reads Port pin when configured as digital input.
0: P1.n pin is logic low.
1: P1.n pin is logic high.

SFR Definition 14.10. P1MDIN: Port1 Input Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xF2

Bits7–0: Analog Input Configuration Bits for P1.7–P1.0 (respectively).
Port pins configured as analog inputs have their weak pullup, digital driver, and digital receiver disabled.
0: Corresponding P1.n pin is configured as an analog input.
1: Corresponding P1.n pin is not configured as an analog input.

SFR Definition 14.11. P1MDOUT: Port1 Output Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xA5

Bits7–0: Output Configuration Bits for P1.7–P1.0 (respectively): ignored if corresponding bit in register P1MDIN is logic 0.
0: Corresponding P1.n Output is open-drain.
1: Corresponding P1.n Output is push-pull.

SFR Definition 14.12. P1SKIP: Port1 Skip

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xD5

Bits7–0: **P1SKIP[7:0]**: Port1 Crossbar Skip Enable Bits.
These bits select Port pins to be skipped by the Crossbar Decoder. Port pins used as analog inputs (for ADC or Comparator) or used as special functions (V_{REF} input, external oscillator circuit, CNVSTR input) should be skipped by the Crossbar.
0: Corresponding P1.n pin is not skipped by the Crossbar.
1: Corresponding P1.n pin is skipped by the Crossbar.

SFR Definition 14.13. P0SKIP: Port0 Skip

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xD4

Bits7–0: **P1SKIP[7:0]**: Port1 Crossbar Skip Enable Bits.
These bits select Port pins to be skipped by the Crossbar Decoder. Port pins used as analog inputs (for ADC or Comparator) or used as special functions (V_{REF} input, external oscillator circuit, CNVSTR input) should be skipped by the Crossbar.
0: Corresponding P1.n pin is not skipped by the Crossbar.
1: Corresponding P1.n pin is skipped by the Crossbar.

SFR Definition 14.14. P1MAT: Port1 Match

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xCF

Bits7–0: **P1MAT[7:0]**: Port1 Match Value.
These bits control the value that unmasked P0 Port pins are compared against. A Port Match event is generated if (P1 & P1MASK) does not equal (P1MAT & P1MASK).

SFR Definition 14.15. P1MASK: Port1 Mask

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xBF

Bits7–0: P1MASK[7:0]: Port1 Mask Value.
These bits select which Port pins will be compared to the value stored in P1MAT.
0: Corresponding P1.n pin is ignored and cannot cause a Port Match event.
1: Corresponding P1.n pin is compared to the corresponding bit in P1MAT.

Table 14.1. Port I/O DC Electrical Characteristics

V_{IO} = 2.7 to 5.25 V, –40 to +125 °C unless otherwise specified

Parameters	Conditions	Min	Typ	Max	Units
Output High Voltage	I_{OH} = –3 mA, Port I/O push-pull	TBD	—	—	V
	I_{OH} = –10 µA, Port I/O push-pull	TBD	—	—	
	I_{OH} = –10 mA, Port I/O push-pull	—	TBD	—	
Output Low Voltage	**V = 2.7 V:**				V
	I_{OL} = 10 µA	—	—	TBD	
	I_{OL} = TBD	—	—	TBD	
	I_{OL} = TBD	—	TBD	—	
	V = 5.25 V:				
	I_{OL} = 10 µA	—	—	TBD	
	I_{OL} = 8.5 mA	—	—	TBD	
	I_{OL} = 25 mA	—	TBD	—	
Input High Voltage		TBD	—	—	V
Input Low Voltage		—	—	TBD	V
Input Leakage Current	Weak Pullup Off	—	—	±TBD	µA
	Weak Pullup On, V_{IN} = 0 V; V = 2.0 V	—	< 0.11	TBD	
	Weak Pullup On, V_{IN} = 0 V; V = 2.4 V	—	< 0.14	TBD	

15. Oscillators

C8051F52x/53x devices include a programmable internal oscillator, an external oscillator drive circuit. The internal oscillator can be enabled/disabled and calibrated using the OSCICN and OSCICL registers, as shown in Figure 15.1. The system clock (SYSCLK) can be derived from the internal oscillator, external oscillator circuit. Oscillator electrical specifications are given in Table 15.1 on page 140.

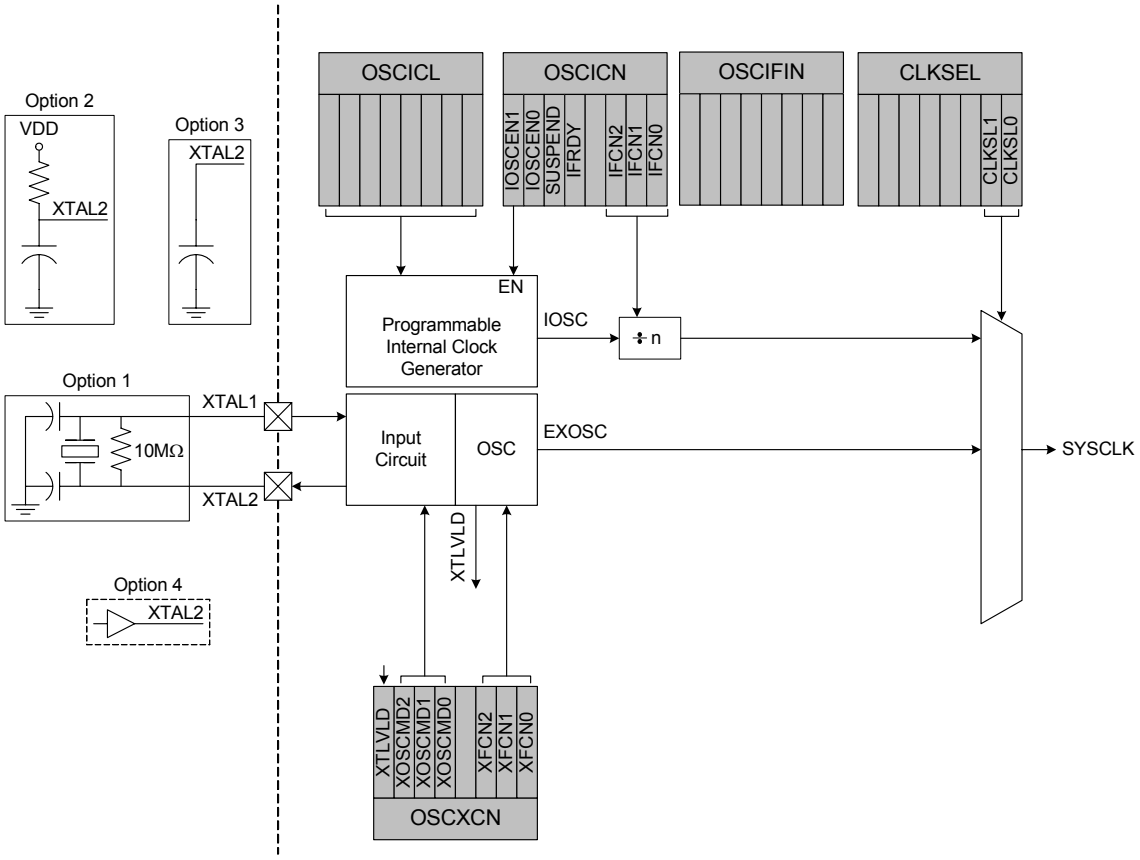


Figure 15.1. Oscillator Diagram

15.1. Programmable Internal Oscillator

All C8051F52x/53x devices include a programmable internal oscillator that defaults as the system clock after a system reset. The internal oscillator period can be programmed via the OSCICL register, shown in SFR Definition 15.2. On C8051F52x/53x devices, OSCICL and OSCIFIN are factory calibrated to obtain a 24.5 MHz frequency.

Electrical specifications for the precision internal oscillator are given in Table 15.1 on page 140. Note that the system clock may be derived from the programmed internal oscillator divided by 1, 2, 4, 8, 16, 32, 64, or 128 as defined by the IFCN bits in register OSCICN. The divide value defaults to 128 following a reset.

15.1.1. Internal Oscillator Suspend Mode

When software writes a logic 1 to SUSPEND (OSCICN.5), the internal oscillator is suspended. If the system clock is derived from the internal oscillator, the input clock to the peripheral or CIP-51 will be stopped until one of the following events occur:

- Port 0 Match Event.
- Port 1 Match Event.
- Comparator 0 enabled and output is logic 0.

When one of the internal oscillator awakening events occur, the internal oscillator, CIP-51, and affected peripherals resume normal operation, regardless of whether the event also causes an interrupt. The CPU resumes execution at the instruction following the write to SUSPEND.

SFR Definition 15.1. OSCICN: Internal Oscillator Control

R/W	R/W	R/W	R	R	R/W	R/W	R/W	Reset Value
IOSCEN1	IOSCEN0	SUSPEND	IFRDY	-	IFCN2	IFCN1	IFCN0	11000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xB2

Bit7–6: **IOSCEN[1:0]**: Internal Oscillator Enable Bits.
00: Oscillator Disabled. Absolute Minimum Power Consumption.
01: Oscillator Enabled in Normal Mode and Disabled in Suspend Mode, Ultra Low Power.
10: Oscillator Enabled in Normal Mode and Disabled in Suspend Mode, Low Power.
11: Oscillator Enabled in Normal Mode and Disabled in Suspend Mode.

Bit5: **SUSPEND**: Internal Oscillator Suspend Enable Bit.
Setting this bit to logic 1 places the internal oscillator in SUSPEND mode. The internal oscillator resumes operation when one of the SUSPEND mode awakening events occur.

Bit4: **IFRDY**: Internal Oscillator Frequency Ready Flag.
0: Internal Oscillator is not running at programmed frequency.
1: Internal Oscillator is running at programmed frequency.

Bits3: **UNUSED**. Read = 00b, Write = don't care.

Bits2–0: **IFCN2–0**: Internal Oscillator Frequency Control Bits.
000: SYSCLK derived from Internal Oscillator divided by 128 (default).
001: SYSCLK derived from Internal Oscillator divided by 64.
010: SYSCLK derived from Internal Oscillator divided by 32.
011: SYSCLK derived from Internal Oscillator divided by 16.
100: SYSCLK derived from Internal Oscillator divided by 8.
101: SYSCLK derived from Internal Oscillator divided by 4.
110: SYSCLK derived from Internal Oscillator divided by 2.
111: SYSCLK derived from Internal Oscillator divided by 1.

SFR Definition 15.2. OSCICL: Internal Oscillator Calibration

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	OSCICL							Varies
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xB3

Bit7: **UNUSED**. Read = 0. Write = don't care.
Bits 6–0: **OSCICL**: Internal Oscillator Calibration Register.
This register determines the internal oscillator period. On C8051F52x/53x devices, the reset value is factory calibrated to generate an internal oscillator frequency of 24.5 MHz.

SFR Definition 15.3. OSCIFIN: Internal Fine Oscillator Calibration

R/W	R/W	R/W	R	R	R/W	R/W	R/W	Reset Value
-	-	OSCIFIN						undetermined
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0xB0

Bit7–6: **UNUSED**. Read = 00b, Write = don't care.
Bit5–0: **OSCIFIN**.

This register is a fine adjustment for the internal oscillator period. On C8051F52x/53x devices, the reset value is factory calibrated to generate an internal oscillator frequency of 24.5 MHz.

15.2. External Oscillator Drive Circuit

The external oscillator circuit may drive an external crystal, ceramic resonator, capacitor, or RC network. A CMOS clock may also provide a clock input. For a crystal or ceramic resonator configuration, the crystal/resonator must be wired across the XTAL1 and XTAL2 pins as shown in Option 1 of Figure 15.1. A 10 MΩ resistor also must be wired across the XTAL1 and XTAL2 pins for the crystal/resonator configuration. In RC, capacitor, or CMOS clock configuration, the clock source should be wired to the XTAL2 pin as shown in Option 2, 3, or 4 of Figure 15.1. The type of external oscillator must be selected in the OSCXCN register, and the frequency control bits (XFCN) must be selected appropriately (see SFR Definition 15.4. OSCXCN: External Oscillator Control).

Important Note on External Oscillator Usage: Port pins must be configured when using the external oscillator circuit. When the external oscillator drive circuit is enabled in crystal/resonator mode, Port pins P0.7 and P1.0 (‘F53x) or P0.2 and P0.3 (‘F52x) are used as XTAL1 and XTAL2 respectively. When the external oscillator drive circuit is enabled in capacitor, RC, or CMOS clock mode, Port pin P1.0 (‘F530) or P0.3 (‘F52x) is used as XTAL2. The Port I/O Crossbar should be configured to skip the Port pins used by the oscillator circuit; see **Section “14.1. Priority Crossbar Decoder” on page 117** for Crossbar configuration. Additionally, when using the external oscillator circuit in crystal/resonator, capacitor, or RC mode, the associated Port pins should be configured as **analog inputs**. In CMOS clock mode, the associated pin should be configured as a **digital input**. See **Section “14.2. Port I/O Initialization” on page 121** for details on Port input mode selection.

15.2.1. Clocking Timers Directly Through the External Oscillator

The external oscillator source divided by eight is a clock option for the timers (**Section “19. Timers” on page 179**) and the Programmable Counter Array (PCA) (**Section “20. Programmable Counter Array (PCA0)” on page 193**). When the external oscillator is used to clock these peripherals, but is not used as the system clock, the external oscillator frequency must be less than or equal to the system clock frequency. In this configuration, the clock supplied to the peripheral (external oscillator / 8) is synchronized with the system clock; the jitter associated with this synchronization is limited to ±0.5 system clock cycles.

15.2.2. External Crystal Example

If a crystal or ceramic resonator is used as an external oscillator source for the MCU, the circuit should be configured as shown in Figure 15.1, Option 1. The External Oscillator Frequency Control value (XFCN) should be chosen from the Crystal column of the table in SFR Definition 15.4. For example, a 12 MHz crystal requires an XFCN setting of 111b.

When the crystal oscillator is first enabled, the oscillator amplitude detection circuit requires a settling time to achieve proper bias. Introducing a delay of 1 ms between enabling the oscillator and checking the XTLVLD bit will prevent a premature switch to the external oscillator as the system clock. Switching to the external oscillator before the crystal oscillator has stabilized can result in unpredictable behavior. The recommended procedure is:

Step 1. Configure XTAL1 and XTAL2 pins by writing ‘1' to the port latch.
Step 2. Configure XTAL1 and XTAL2 as analog inputs.
Step 3. Enable the external oscillator.
Step 4. Wait at least 1 ms.
Step 5. Poll for XTLVLD => '1'.
Step 6. Switch the system clock to the external oscillator.

Note: Tuning-fork crystals may require additional settling time before XTLVLD returns a valid result.

The capacitors shown in the external crystal configuration provide the load capacitance required by the crystal for correct oscillation. These capacitors are "in series" as seen by the crystal and "in parallel" with the stray capacitance of the XTAL1 and XTAL2 pins.

Note: The load capacitance depends upon the crystal and the manufacturer. Please refer to the crystal data sheet when completing these calculations.

For example, a tuning-fork crystal of 32.768 Hz with a recommended load capacitance of 12.5 pF should use the configuration shown in Figure 15.1, Option 1. The total value of the capacitors and the stray capacitance of the XTAL pins should equal 25 pF. With a stray capacitance of 3 pF per pin, the 22 pF capacitors yield an equivalent capacitance of 12.5 pF across the crystal, as shown in Figure 15.2.

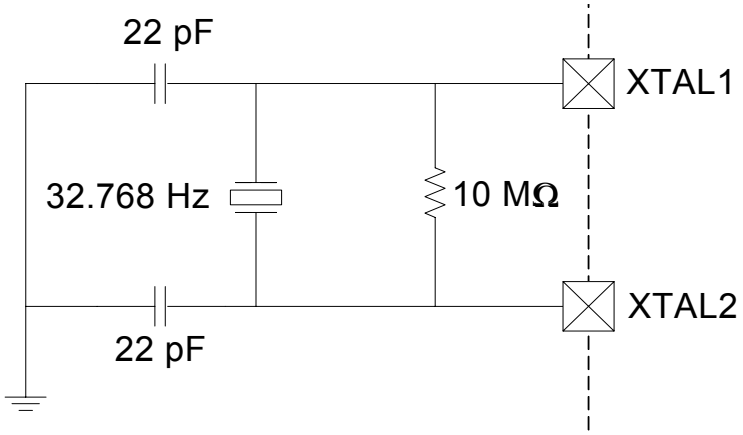


Figure 15.2. 32.768 Hz External Crystal Example

Important Note on External Crystals: Crystal oscillator circuits are quite sensitive to PCB layout. The crystal should be placed as close as possible to the XTAL pins on the device. The traces should be as short as possible and shielded with ground plane from any other traces which could introduce noise or interference.

15.2.3. External RC Example

If an RC network is used as an external oscillator source for the MCU, the circuit should be configured as shown in Figure 15.1, Option 2. The capacitor should be no greater than 100 pF; however for very small capacitors, the total capacitance may be dominated by parasitic capacitance in the PCB layout. To determine the required External Oscillator Frequency Control value (XFCN) in the OSCXCN Register, first select the RC network value to produce the desired frequency of oscillation. If the frequency desired is 100 kHz, let R = 246 kΩ and C = 50 pF:

f = 1.23(10^3) / RC = 1.23 (10^3) / [246 x 50] = 0.1 MHz = 100 kHz

Referring to the table in SFR Definition 15.4, the required XFCN setting is 010b. Programming XFCN to a higher setting in RC mode will improve frequency accuracy at a slightly increased external oscillator supply current.

15.2.4. External Capacitor Example

If a capacitor is used as an external oscillator for the MCU, the circuit should be configured as shown in Figure 15.1, Option 3. The capacitor should be no greater than 100 pF; however for very small capacitors, the total capacitance may be dominated by parasitic capacitance in the PCB layout. To determine the required External Oscillator Frequency Control value (XFCN) in the OSCXCN Register, select the frequency of oscillation and calculate the capacitance to be used from the equations below. Assume V_{DD} = 2.0 V and f = 75 kHz:

f = KF / (C x V_{DD})
0.075 MHz = KF / (C x 2.0)

Since the frequency of roughly 75 kHz is desired, select the K Factor from the table in SFR Definition 15.4 as KF = 7.7:

0.075 MHz = 7.7 / (C x 2.0)

C x 2.0 = 7.7 / 0.075 MHz

C = 102.6 / 2.0 pF = 51.3 pF

Therefore, the XFCN value to use in this example is 010b.

SFR Definition 15.4. OSCXCN: External Oscillator Control

R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
XTLVLD	XOSCMD2	XOSCMD1	XOSCMD0	Reserved	XFCN2	XFCN1	XFCN0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xB1

Bit7: **XTLVLD**: Crystal Oscillator Valid Flag. (Read only when XOSCMD = 11x.)
0: Crystal Oscillator is unused or not yet stable.
1: Crystal Oscillator is running and stable.

Bits6–4: **XOSCMD2–0**: External Oscillator Mode Bits.
00x: External Oscillator circuit off.
010: External CMOS Clock Mode.
011: External CMOS Clock Mode with divide by 2 stage.
100: RC Oscillator Mode.
101: Capacitor Oscillator Mode.
110: Crystal Oscillator Mode.
111: Crystal Oscillator Mode with divide by 2 stage.

Bit3: **RESERVED**. Read = 0b; Must write 0b.

Bits2–0: **XFCN2–0**: External Oscillator Frequency Control Bits.
000-111: See table below:

XFCN	Crystal (XOSCMD = 11x)	RC (XOSCMD = 10x)	C (XOSCMD = 10x)
000	$f \leq 20$ kHz	$f \leq 25$ kHz	K Factor = 0.87
001	20 kHz $< f \leq$ 58 kHz	25 kHz $< f \leq$ 50 kHz	K Factor = 2.6
010	58 kHz $< f \leq$ 155 kHz	50 kHz $< f \leq$ 100 kHz	K Factor = 7.7
011	155 kHz $< f \leq$ 415 kHz	100 kHz $< f \leq$ 200 kHz	K Factor = 22
100	415 kHz $< f \leq$ 1.1 MHz	200 kHz $< f \leq$ 400 kHz	K Factor = 65
101	1.1 MHz $< f \leq$ 3.1 MHz	400 kHz $< f \leq$ 800 kHz	K Factor = 180
110	3.1 MHz $< f \leq$ 8.2 MHz	800 kHz $< f \leq$ 1.6 MHz	K Factor = 664
111	8.2 MHz $< f \leq$ 25 MHz	1.6 MHz $< f \leq$ 3.2 MHz	K Factor = 1590

Crystal Mode (Circuit from Figure 15.1, Option 1; XOSCMD = 11x)
Choose XFCN value to match crystal or resonator frequency.

RC Mode (Circuit from Figure 15.1, Option 2; XOSCMD = 10x)
Choose XFCN value to match frequency range:
$f = 1.23(10^3) / (R \times C)$, where
f = frequency of clock in MHz
C = capacitor value in pF
R = Pullup resistor value in kΩ

C Mode (Circuit from Figure 15.1, Option 3; XOSCMD = 10x)
Choose K Factor (KF) for the oscillation frequency desired:
$f = KF / (C \times V_{DD})$, where
f = frequency of clock in MHz
C = capacitor value the XTAL2 pin in pF
V_{DD} = Power Supply on MCU in volts

15.3. System Clock Selection

The internal oscillator requires little start-up time and may be selected as the system clock immediately following the OSCICN write that enables the internal oscillator. External crystals and ceramic resonators typically require a start-up time before they are settled and ready for use. The Crystal Valid Flag (XTLVLD in register OSCXCN) is set to '1' by hardware when the external oscillator is settled. **To avoid reading a false XTLVLD, in crystal mode software should delay at least 1 ms between enabling the external oscillator and checking XTLVLD.** RC and C modes typically require no startup time.

The CLKSL bit in register CLKSEL selects which oscillator source is used as the system clock. CLKSL must be set to '1' for the system clock to run from the external oscillator; however the external oscillator may still clock certain peripherals (timers, PCA) when another oscillator is selected as the system clock. The system clock may be switched on-the-fly between the internal oscillator and external oscillator, as long as the selected clock source is enabled and has settled.

SFR Definition 15.5. CLKSEL: Clock Select

R	R	R/W	R/W	R	R/W	R/W	R/W	Reset Value
-	-	Reserved	Reserved	-	Reserved	Reserved	CLKSL	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xA9

Bits7–6: Unused. Read = 00b; Write = don't care.
Bits5–4: Reserved. Read = 0b; Must write 0b.
Bit3: Unused. Read = 0b; Write = don't care.
Bits2–1: Reserved. Read = 0b; Must write 0b.
Bit0: **CLKSL**: System Clock Select
0: Internal Oscillator (as determined by the IFCN bits in register OSCICN).
1: External Oscillator.

Table 15.1. Oscillator Electrical Characteristics

–40 to +125 °C unless otherwise specified

Parameter	Conditions	Min	Typ	Max	Units
Internal Oscillator Frequency	Reset Frequency	24	24.5	25	MHz
Internal Oscillator OFF Supply Current	OSCICN.7 = 0 OSCICN.6 = 0	—	—	—	µA
Internal Oscillator ON (Ultra Low Power Mode) Supply Current	OSCICN.7 = 0 OSCICN.6 = 1	—	—	—	µA
Internal Oscillator ON (Very Low Power Mode) Supply Current	OSCICN.7 = 1 OSCICN.6 = 0	—	—	—	µA
Internal Oscillator ON (Low Power Mode) Supply Current	OSCICN.7 = 1 OSCICN.6 = 1	—	—	—	µA

16. UART0

UART0 is an asynchronous, full duplex serial port offering modes 1 and 3 of the standard 8051 UART. Enhanced baud rate support allows a wide range of clock sources to generate standard baud rates (details in **Section "16.1. Enhanced Baud Rate Generation" on page 142**). Received data buffering allows UART0 to start reception of a second incoming data byte before software has finished reading the previous data byte. (Please refer to **Section "21. Revision Specific Behavior" on page 209** for more information on the pins associated with the UART interface.)

UART0 has two associated SFRs: Serial Control Register 0 (SCON0) and Serial Data Buffer 0 (SBUF0). The single SBUF0 location provides access to both transmit and receive registers. **Writes to SBUF0 always access the Transmit register. Reads of SBUF0 always access the buffered Receive register; it is not possible to read data from the Transmit register.**

With UART0 interrupts enabled, an interrupt is generated each time a transmit is completed (TI0 is set in SCON0), or a data byte has been received (RI0 is set in SCON0). The UART0 interrupt flags are not cleared by hardware when the CPU vectors to the interrupt service routine. They must be cleared manually by software, allowing software to determine the cause of the UART0 interrupt (transmit complete or receive complete).

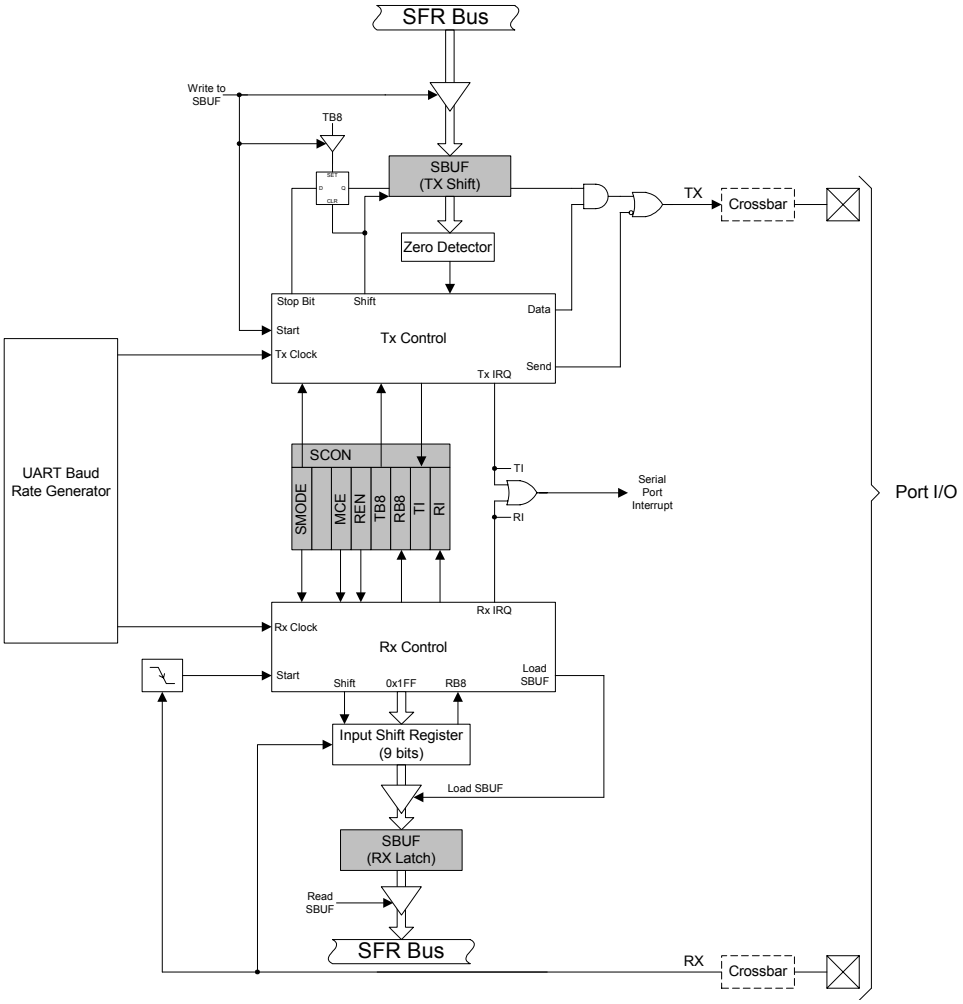


Figure 16.1. UART0 Block Diagram

16.1. Enhanced Baud Rate Generation

The UART0 baud rate is generated by Timer 1 in 8-bit auto-reload mode. The TX clock is generated by TL1; the RX clock is generated by a copy of TL1 (shown as RX Timer in Figure 16.2), which is not user-accessible. Both TX and RX Timer overflows are divided by two to generate the TX and RX baud rates. The RX Timer runs when Timer 1 is enabled, and uses the same reload value (TH1). However, an RX Timer reload is forced when a START condition is detected on the RX pin. This allows a receive to begin any time a START is detected, independent of the TX Timer state.

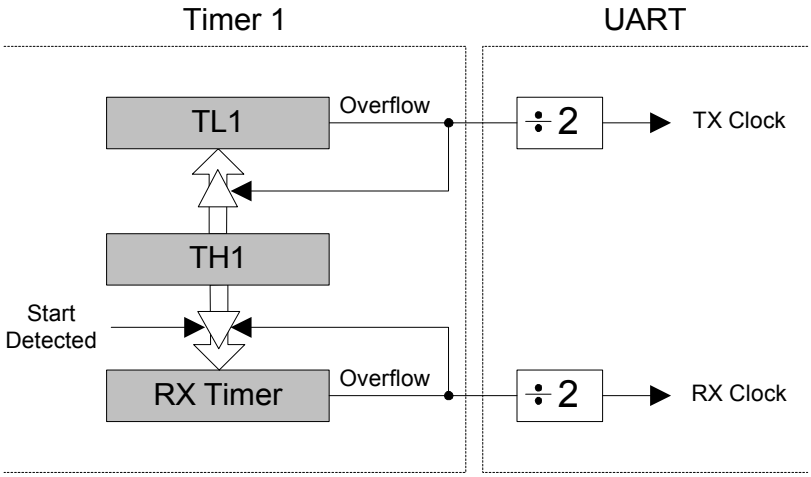


Figure 16.2. UART0 Baud Rate Logic

Timer 1 should be configured for Mode 2, 8-bit auto-reload (see **Section "19.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload" on page 181**). The Timer 1 reload value should be set so that overflows will occur at two times the desired UART baud rate frequency. Note that Timer 1 may be clocked by one of six sources: SYSCLK, SYSCLK / 4, SYSCLK / 12, SYSCLK / 48, the external oscillator clock / 8, or an external input T1. The UART0 baud rate is determined by Equation 16.1-A and Equation 16.1-B.

$$\text{A)}\quad \text{UartBaudRate} = \frac{1}{2} \times \text{T1\_Overflow\_Rate}$$

$$\text{B)}\quad \text{T1\_Overflow\_Rate} = \frac{\text{T1}_{CLK}}{256 - \text{TH1}}$$

Equation 16.1. UART0 Baud Rate

Where $T1_{CLK}$ is the frequency of the clock supplied to Timer 1, and *T1H* is the high byte of Timer 1 (8-bit auto-reload mode reload value). Timer 1 clock frequency is selected as described in **Section "19. Timers" on page 179**. A quick reference for typical baud rates and system clock frequencies is given in Table 16.1. Note that the internal oscillator may still generate the system clock when the external oscillator is driving Timer 1.

16.2. Operational Modes

UART0 provides standard asynchronous, full duplex communication. The UART mode (8-bit or 9-bit) is selected by the S0MODE bit (SCON0.7). Typical UART connection options are shown below.

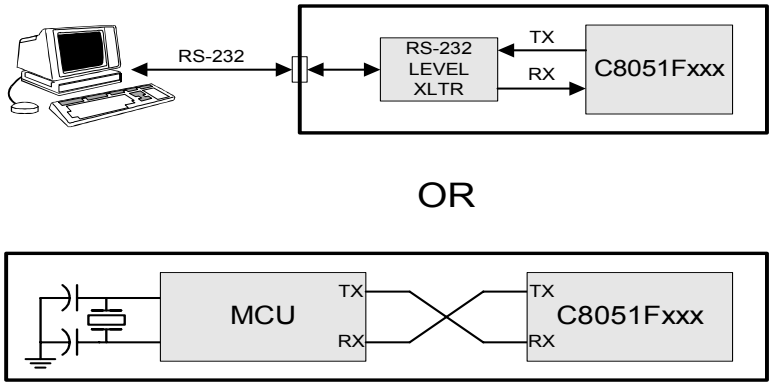


Figure 16.3. UART Interconnect Diagram

16.2.1. 8-Bit UART

8-Bit UART mode uses a total of 10 bits per data byte: one start bit, eight data bits (LSB first), and one stop bit. Data are transmitted LSB first from the TX0 pin and received at the RX0 pin. On receive, the eight data bits are stored in SBUF0 and the stop bit goes into RB80 (SCON0.2).

Data transmission begins when software writes a data byte to the SBUF0 register. The TI0 Transmit Interrupt Flag (SCON0.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN0 Receive Enable bit (SCON0.4) is set to logic 1. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: RI0 must be logic 0, and if MCE0 is logic 1, the stop bit must be logic 1. In the event of a receive data overrun, the first received 8 bits are latched into the SBUF0 receive register and the following overrun data bits are lost.

If these conditions are met, the eight bits of data is stored in SBUF0, the stop bit is stored in RB80 and the RI0 flag is set. If these conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set. An interrupt will occur if enabled when either TI0 or RI0 is set.

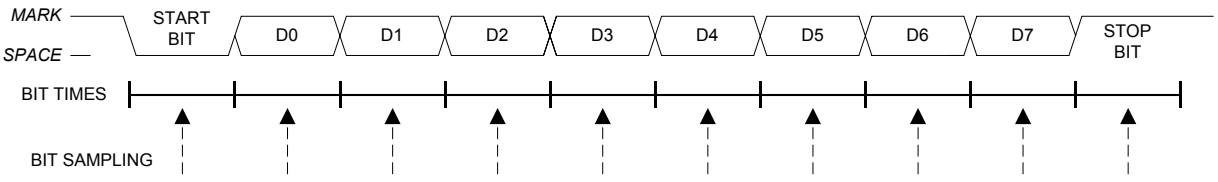


Figure 16.4. 8-Bit UART Timing Diagram

16.2.2. 9-Bit UART

9-bit UART mode uses a total of eleven bits per data byte: a start bit, 8 data bits (LSB first), a programmable ninth data bit, and a stop bit. The state of the ninth transmit data bit is determined by the value in TB80 (SCON0.3), which is assigned by user software. It can be assigned the value of the parity flag (bit P in register PSW) for error detection, or used in multiprocessor communications. On receive, the ninth data bit goes into RB80 (SCON0.2) and the stop bit is ignored.

Data transmission begins when an instruction writes a data byte to the SBUF0 register. The TI0 Transmit Interrupt Flag (SCON0.1) is set at the end of the transmission (the beginning of the stop-bit time). Data reception can begin any time after the REN0 Receive Enable bit (SCON0.4) is set to '1'. After the stop bit is received, the data byte will be loaded into the SBUF0 receive register if the following conditions are met: (1) RI0 must be logic 0, and (2) if MCE0 is logic 1, the 9th bit must be logic 1 (when MCE0 is logic 0, the state of the ninth data bit is unimportant). If these conditions are met, the eight bits of data are stored in SBUF0, the ninth bit is stored in RB80, and the RI0 flag is set to '1'. If the above conditions are not met, SBUF0 and RB80 will not be loaded and the RI0 flag will not be set to '1'. A UART0 interrupt will occur if enabled when either TI0 or RI0 is set to '1'.

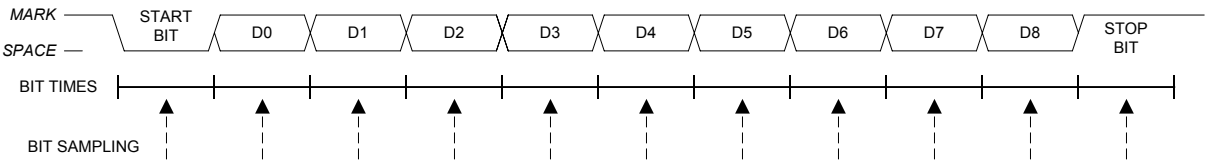


Figure 16.5. 9-Bit UART Timing Diagram

16.3. Multiprocessor Communications

9-Bit UART mode supports multiprocessor communication between a master processor and one or more slave processors by special use of the ninth data bit. When a master processor wants to transmit to one or more slaves, it first sends an address byte to select the target(s). An address byte differs from a data byte in that its ninth bit is logic 1; in a data byte, the ninth bit is always set to logic 0.

Setting the MCE0 bit (SCON0.5) of a slave processor configures its UART such that when a stop bit is received, the UART will generate an interrupt only if the ninth bit is logic 1 (RB80 = 1) signifying an address byte has been received. In the UART interrupt handler, software will compare the received address with the slave's own assigned 8-bit address. If the addresses match, the slave will clear its MCE0 bit to enable interrupts on the reception of the following data byte(s). Slaves that weren't addressed leave their MCE0 bits set and do not generate interrupts on the reception of the following data bytes, thereby ignoring the data. Once the entire message is received, the addressed slave resets its MCE0 bit to ignore all transmissions until it receives the next address byte.

Multiple addresses can be assigned to a single slave and/or a single address can be assigned to multiple slaves, thereby enabling "broadcast" transmissions to more than one slave simultaneously. The master processor can be configured to receive all transmissions or a protocol can be implemented such that the master/slave role is temporarily reversed to enable half-duplex transmission between the original master and slave(s).

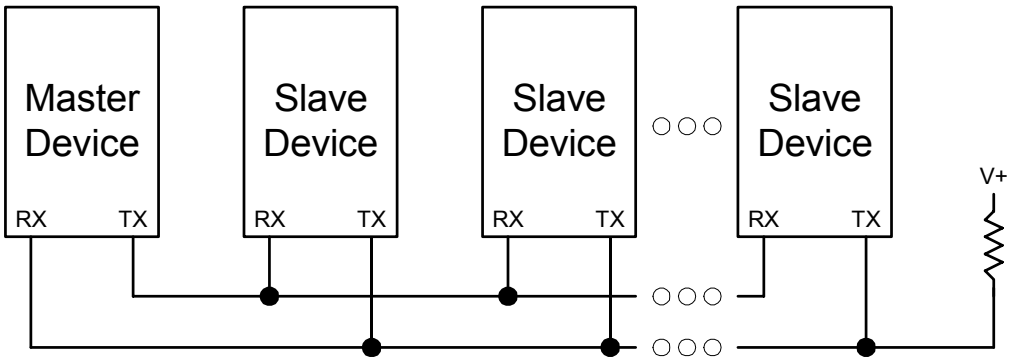


Figure 16.6. UART Multi-Processor Mode Interconnect Diagram

SFR Definition 16.1. SCON0: Serial Port 0 Control

R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
S0MODE	-	MCE0	REN0	TB80	RB80	TI0	RI0	01000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0x98

Bit7: **S0MODE**: Serial Port 0 Operation Mode.
This bit selects the UART0 Operation Mode.
0: 8-bit UART with Variable Baud Rate.
1: 9-bit UART with Variable Baud Rate.

Bit6: **UNUSED**. Read = 1b. Write = don't care.

Bit5: **MCE0:** Multiprocessor Communication Enable.
The function of this bit is dependent on the Serial Port 0 Operation Mode.
S0MODE = 0: Checks for valid stop bit.
0: Logic level of stop bit is ignored.
1: RI0 will only be activated if stop bit is logic level 1.
S0MODE = 1: Multiprocessor Communications Enable.
0: Logic level of ninth bit is ignored.
1: RI0 is set and an interrupt is generated only when the ninth bit is logic 1.

Bit4: **REN0:** Receive Enable.
This bit enables/disables the UART receiver.
0: UART0 reception disabled.
1: UART0 reception enabled.

Bit3: **TB80**: Ninth Transmission Bit.
The logic level of this bit will be assigned to the ninth transmission bit in 9-bit UART Mode. It is not used in 8-bit UART Mode. Set or cleared by software as required.

Bit2: **RB80**: Ninth Receive Bit.
RB80 is assigned the value of the STOP bit in Mode 0; it is assigned the value of the 9th data bit in Mode 1.

Bit1: **TI0**: Transmit Interrupt Flag.
Set by hardware when a byte of data has been transmitted by UART0 (after the 8th bit in 8-bit UART Mode, or at the beginning of the STOP bit in 9-bit UART Mode). When the UART0 interrupt is enabled, setting this bit causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.

Bit0: **RI0**: Receive Interrupt Flag.
Set to '1' by hardware when a byte of data has been received by UART0 (set at the STOP bit sampling time). When the UART0 interrupt is enabled, setting this bit to '1' causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.

SFR Definition 16.2. SBUF0: Serial (UART0) Port Data Buffer

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x99

Bits7–0: **SBUF0[7:0]**: Serial Data Buffer Bits 7–0 (MSB–LSB)
This SFR accesses two registers; a transmit shift register and a receive latch register. When data is written to SBUF0, it goes to the transmit shift register and is held for serial transmission. Writing a byte to SBUF0 initiates the transmission. A read of SBUF0 returns the contents of the receive latch.

Table 16.1. Timer Settings for Standard Baud Rates Using the Internal Oscillator

	Frequency: 24.5 MHz						
	Target Baud Rate (bps)	Baud Rate % Error	Oscilla-tor Divide Factor	Timer Clock Source	SCA1–SCA0 (pre-scale select)*	T1M*	Timer 1 Reload Value (hex)
	230400	–0.32%	106	SYSCLK	XX	1	0xCB
	115200	–0.32%	212	SYSCLK	XX	1	0x96
	57600	0.15%	426	SYSCLK	XX	1	0x2B
	28800	–0.32%	848	SYSCLK / 4	01	0	0x96
	14400	0.15%	1704	SYSCLK / 12	00	0	0xB9
	9600	–0.32%	2544	SYSCLK / 12	00	0	0x96
	2400	–0.32%	10176	SYSCLK / 48	10	0	0x96
	1200	0.15%	20448	SYSCLK / 48	10	0	0x2B

X = Don’t care

***Note:** SCA1–SCA0 and T1M bit definitions can be found in Section 19.1.

17. LIN (C8051F520/523/526/530/533/536 only)

LIN is an asynchronous, serial communications interface used primarily in automotive networks. This document assumes previous knowledge of the interface. For more information about the LIN concept including specifications please refer to the LIN consortium (http://www.lin-subbus.org/).

17.1. Major Characteristics

Silicon Laboratories LIN peripheral implements a complete LIN interface and presents the following features:

1. Selectable Master and Slave Modes
2. Unique Self-Synchronization without a quartz crystal or ceramic resonator in both Master and Slave modes. Silicon Laboratories innovative internal oscillator design technology allows the oscillator to reach 0.5% precision across the entire supply voltage and temperature range.
3. Fully configurable transmission/reception characteristics via SFRs
   Important: The minimum system clock (SYSCLK) to be used when using the LIN peripheral is of 8 MHz.

The following Figure describes LIN main blocks.

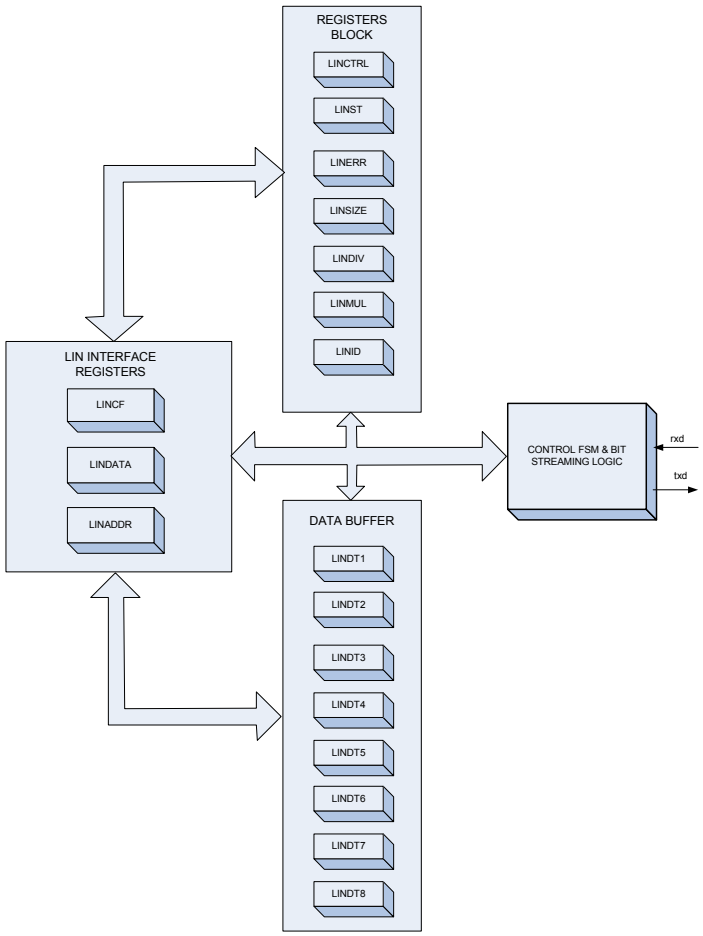


Figure 17.1. LIN Flowchart

The LIN peripheral is made of four major logic groups:

- LIN Interface Registers - Provide the interface between the microcontroller core and the peripheral.
- Data Buffer - Contains the registers where transmitted and received message data bytes are placed
- Registers Block - Contain all registers used to control the functionality of the interface
- Control State Machine and Bit Streaming Logic - Contains the hardware the serializes messages and the timing control of the peripheral.

The LIN module does not directly support LIN Version 1.3 Extended Frames. In the case of a slave configuration if the application detects an extended frame it has to write a ‘1’ to the **STOP** bit (**LINCTRL**) instead of setting the **DTACK** bit (steps 1b...1e can then be skipped). In that case the LIN peripheral stops the processing of the LIN communication until the next SYNC BREAK is received.

17.2. Software Interface with the LIN Peripheral

The communication with the LIN interface is done indirectly through a pair of registers called LINADDR and LINDATA and the Selection of the mode (Master or Slave) and the automatic baud rate feature are done though the LINCF register. To write into a specific register other than these three ones require the user to first load the LINADDR register with the address of the required LIN register and then load the data to be transferred to the register using LINDATA.

17.3. LIN Registers

The following Special Function Registers (SFRs) are available:

17.3.1. LIN Direct Access SFR Registers Definition

SFR Definition 17.1. LINADDR: Indirect Address Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x92

Bit7–0: **LINADDR[7:0]** LIN address Register

SFR Definition 17.2. LINDATA: LIN Data Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x93

Bit7–0: **LINDATA[7:0]** LIN Data Register

SFR Definition 17.3. LINCF Control Mode Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
MODE	ABAUD							00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x95

Bit7: **MODE:** LIN Mode Selection

1 - Master Mode
0 - Slave Mode

Bit6: **ABAUD:** LIN Mode Automatic Bit Rate Selection (slave mode only)

1 - Automatic bit-rate Selection
0 - Manual bit-rate Selection

17.3.2. LIN Indirect Access SFR Registers Definition

Table 17.1. LIN Registers* (Indirectly Addressable)

Name	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LINDT1	**0x00**	data byte 0[7:0]							
LINDT2	**0x01**	data byte 1[7:0]							
LINDT3	**0x02**	data byte 2[7:0]							
LINDT4	**0x03**	data byte 3[7:0]							
LINDT5	**0x04**	data byte 4[7:0]							
LINDT6	**0x05**	data byte 5[7:0]							
LINDT7	**0x06**	data byte 6[7:0]							
LINDT8	**0x07**	data byte 7[7:0]							
LINCTRL	**0x08**	STOP(s)	SLEEP(s)	TXRX	DTACK(s)	RSTINT	RSTERR	WUPREQ	STREQ(m)
LINST	**0x09**	ACTIVE	IDLTOUT	ABORT(s)	DTREQ(s)	LININT	ERROR	WAKEUP	DONE
LINERR	**0x0A**				SYNCH(s)	PRTY(s)	TOUT	CHK	BITERR
LINSIZE	**0x0B**	ENHCHK				data length [3:0]			
LINDIV	**0x0C**	baud divider[7:0]							
LINMUL	**0x0D**	PRESCL1	PRESCL0	MUL4	MUL3	MUL2	MUL1	MUL0	DIV
LINID	**0x0E**			ID5	ID4	ID3	ID2	ID1	ID0

***These registers are used in both master and slave mode. The register bits marked with (m) are accessible only in Master mode while the register bits marked with (s) are accessible only in slave mode. All other registers are accessible in both modes.**

SFR Definition 17.4. LINDT1: LIN Data Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x00 (indirect)

Bit7–0: **LINDT1**: Serial Data Byte Received or transmitted by the interface

SFR Definition 17.5. LINDT2: LIN Data Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x01 (indirect)

Bit7–0: **LINDT2**: Serial Data Byte Received or transmitted by the interface

SFR Definition 17.6. LINDT3: LIN Data Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x02 (indirect)

Bit7–0: **LINDT3**: Serial Data Byte Received or transmitted by the interface

SFR Definition 17.7. LINDT4: LIN Data Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x03 (indirect)

Bit7–0: **LINDT4**: Serial Data Byte Received or transmitted by the interface

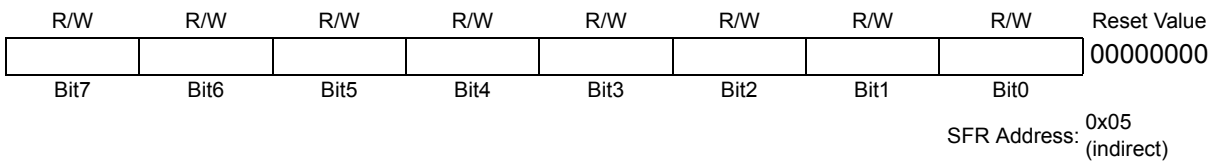
SFR Definition 17.8. LINDT5: LIN Data Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x04 (indirect)

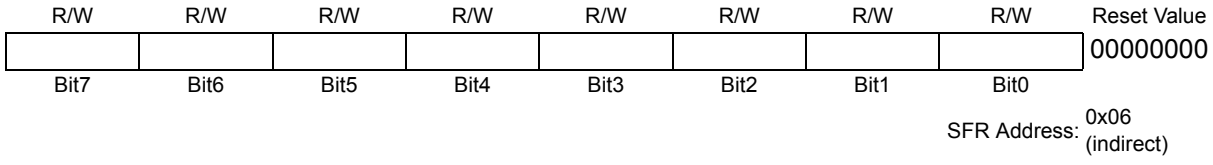
Bit7–0: **LINDT5**: Serial Data Byte Received or transmitted by the interface

SFR Definition 17.9. LINDT6: LIN Data Byte



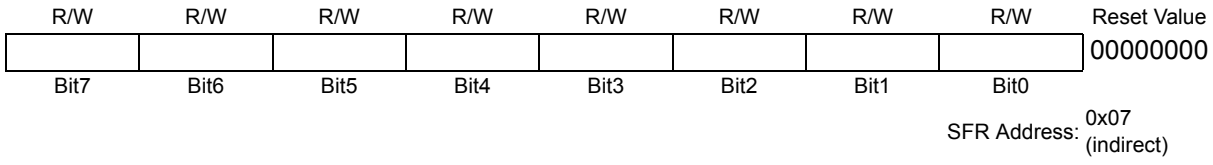
Bit7–0: **LINDT6**: Serial Data Byte Received or transmitted by the interface

SFR Definition 17.10. LINDT7: LIN Data Byte



Bit7–0: **LINDT7**: Serial Data Byte Received or transmitted by the interface

SFR Definition 17.11. LINDT8: LIN Data Byte



Bit7–0: **LINDT8**: Serial Data Byte Received or transmitted by the interface

SFR Definition 17.12. LINCTRL: LIN Control Register

W	W	W	R/W	R/W	R/W	R/W	R/W	Reset Value
STOP	SLEEP	TXRX	DTACK	RSTINT	RSTERR	WUPREQ	STREQ	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x08 (indirect)

Bit7: **STOP**: Processing of the LIN communications Blocking.(slave mode only)
This bit is to be set by the application to block the processing of the LIN Communications until the next SYNCH BREAK signal. It is used when the application is handling a data request interrupt and cannot use the frame content with the received identifier (always reads '0').

Bit6: **SLEEP**: Sleep Mode Warning.
This bit is to be set by the application to warn the peripheral that a Sleep Mode Frame was received and that the Bus is in sleep mode or if a Bus Idle timeout interrupt is requested. The application must reset it when a Wake-Up interrupt is requested.

Bit5: **TXRX**: Transmit/Receive Selection Bit.
This bit is to be set by the application to select if the current frame is a transmit frame or a receive frame.
1 - Transmit Operation
0 - Receive Operation

Bit4: **DTACK**: Data acknowledge bit.(slave mode only)
This bit is to be set by the application after handling a data request interrupt (reset by the peripheral.)

Bit3: **RSTINT**: Interrupt Reset bit.
This bit must be set by the application to reset the "INT" bit in the LINST (LIN status register).

Bit2: **RSTERR**: Error Reset Bit
The application must set this bit in order to reset the error bits in the LINST (LIN status Register) and the LINERR (LIN Error Register) bits.

Bit1: **WUPREQ**: Wake-Up Request Bit.
This bit must be set by the application to end the sleep mode of the LIN bus by sending a Wake-Up Signal. (reset by the peripheral)

Bit0: **STREQ**: Start Request Bit.(master mode only)
This bit must be set by the application to start a LIN transmission. It may be done only after loading the identifier, data length and data buffer.(reset by the peripheral upon completion of the transmission or error detection).

SFR Definition 17.13. LINST: LIN STATUS Register

R	R	R	R	R/W	R	R	R	Reset Value
ACTIVE	IDLTOUT	ABORT	DTREQ	LININT	ERROR	WAKEUP	DONE	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x09 (indirect)

Bit7: **ACTIVE**: LIN Bus Activity Bit.
This bit shows when transmission activity in the bus is detected by the peripheral.
1 - Transmission activity detected by the peripheral
0 - No transmission detected by the peripheral.

Bit6: **IDLTOUT**: Bus Idle Timeout.(slave mode only)
This bit is set by the peripheral if no bus activity is detected over a period of 4 seconds and the SLEEP bit in the LINCTRL (LIN Control Register) is not set by the application. Upon setting this bit the peripheral also set the interrupt Request Bit (LININT) and the applications can then assume that the LIN bus is in sleep mode and set the SLEEP bit.

Bit5: **ABORT**: Aborted transmission signal.(slave mode only)
This bit is set by the peripheral when a new SYNCH BREAK signal is detected before the end of the last transmission. The transmission is aborted and the new frame is processed. This bit is also set when the application sets the STOP bit (from LINCTRL). Once a SYNCH BREAK signal is received (if it doesn't interrupt another transmission) this signal is reset.

Bit4: **DTREQ**: Data Request bit.(slave mode only)
The peripheral sets this bit after receiving the Identifier and requests an interrupt. The application has then to perform the following actions:
1- Decode the Identifier to decide whether the current frame is a transmit or a receive operation.
2- Adjust the TXRX bit (in LINCTRL) and to load the data length.
3- In case of a transmit operation the application must load the data buffer.
4- Set the DTACK bit (data acknowledge) found in the LINCTRL register.

Bit3: **LININT**: Interrupt Request bit.
This bit is set when an interrupt is issued and has to be reset by the application by setting the RSTINT bit (LINCTRL)

Bit2: **ERROR**: Communication Error Bit.
The peripheral sets the bit if an error has been detected. The bit has to be reset by the application by setting the RSTERR bit (LINCTRL).

Bit1: **WAKEUP**: Wake-Up Request Bit.
The bit is set when the peripheral is transmitting a Wake-Up signal or has received a Wake-Up signal.

Bit0: **DONE**: Transmission Complete Bit.
The peripheral sets this bit at the end of a successful transmission and resets it at the start of another transmission.

SFR Definition 17.14. LINERR: LIN ERROR Register

R	R	R	R	R	R	R	R	Reset Value
			SYNCH	PRTY	TOUT	CHK	BITERR	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x0A (indirect)

Bit7–5: **UNUSED**. Read = 000b. Write = don't care.

Bit4: **SYNCH**: Synchronization Error bit.(slave mode only)
The peripheral detected edges of the SYNCH FIELD outside the maximum tolerance.

Bit3: **PRTY**: Parity Error bit.(slave mode only)
This bit is set when a parity error is detected.

Bit2: **TOUT**: Timeout Error Bit.
This bit is set whenever one of the following conditions is met:
1- The master detects a timeout error if it is expecting data from the bus but no slave does respond.
2- If the slave responds to late and the frame is not finished within the maximum frame length T_{FRAME_MAX}.
3- The slave detects a timeout error if it is expecting data from the master or another slave but no data is transmitted on the bus.
4- If the frame is not finished within the maximum frame length T^{FRAME_MAX} is reached.
5- The slave detects a timeout error if it is requesting a data acknowledge to the application (for selecting receive or transmit, data length and loading data) and the application does not set the DTACK bit (LINCTRL) or STOP bit (LINCTRL) until the end of the reception of the first byte after the identifier.
6- The slave detects a timeout error if it has transmitted a wakeup signal and it detects no sync field (from the master) within 150 ms.

Bit1: **CHK**: Checksum Error Bit.
The bit is set when the peripheral detects a checksum error.

Bit0: **BITERR**: Bit Error bit.
This error bit is set when the bit value monitored by the peripheral is different from the one sent.

SFR Definition 17.15. LINSIZE: LIN Message Size Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
ENHCHK	-	-	-	LINSIZE3	LINSIZE2	LINSIZE1	LINSIZE0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x0B (indirect)

Bit7: **ENHCHK**: Checksum version selection bit
1 - Spec. 2.0, inverted eight bit sum with carry over all data bytes and protected identifier.
0 - Spec 1.3, inverted eight bit sum with carry over all data bytes.

Bit6–4: **UNUSED**. Read = 00b. Write = don't care.

Bit3–0: **LINSIZE3–0**: LIN data field size.
If the LINSIZE bits are filled ("1111b') then the size of the data field is defined as a function of the two most significative bits of the identifier as defined in the table below, otherwise is defined by the LINSIZE3–0 bits.

Table 17.2. Number of Data Bytes

ID5	ID4	Number of Bytes in the Data Field
0	0	2 bytes
0	1	2 bytes
1	0	4 bytes
1	1	8 bytes

Bit0: **UNUSED**. Read = 00b. Write = don't care.

SFR Definition 17.16. LINDIV: LIN Divider Register

R	R	R	R	R	R	R	R	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x0C (indirect)

Bit7–0: **Baud Rate Divider [7:0]**.
This register contains the 8 least significative bits of the divider used to generate the baud rate.

SFR Definition 17.17. LINMUL: LIN Multiplier Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
PRESCL1	PRESCL0	MUL4	MUL3	MUL2	MUL1	MUL0	DIV	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x0D (indirect)

Bit7–6: **PRESCL1–0**:Prescaler used to create the baud rate.
Bit5–1: **MUL4–0**: Multiplier used to create the baud rate.
Bit0: **DIV**: Most significative bit of the divider used to create the baud rate.

SFR Definition 17.18. LINID: LIN ID Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
		ID5	ID4	ID3	ID2	ID1	ID0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x0E (indirect)

Bit7–6: **UNUSED**. Read = 00b. Write = don't care.
Bit5–0: **ID5–0**: Identifier

17.4. Bit Timing

The LIN bit rate is based on the SYSCLK and the Bit-Timing Registers according to the following equation:

f_{bit} = SYSCLK (MHz) / ($2^{(prescaler + 1)}$ x divider x (multiplier+1))

The divider factor must be in the range of 200 to 511 and the SYSCLK must be at least of 8 MHz in order to achieve the tolerances required by the LIN protocol. Also the procedure to adjust the Bit-Timing registers is different between master mode, slave mode with automatic bit-rate detection and slave without bit-rate detection.

17.4.1. Bit Timing Register adjustment of Master and Slave without Automatic Bit-Rate Detection

The necessary steps to adjust the Bit-Timing registers of a master or slave without automatic bit-rate detection are:

1. Set up the bit time multiplier depending on used LIN data rate f_{Bit} according to the following equation:
   (the result multiplier value must be rounded down to the nearest integer)

$$multiplier = ((20\ kB/s)/f_{Bit}) - 1$$

2. Set up the prescaler register depending on system clock, data rate and bit time multiplier according to the following equation:
   (the result prescaler value must be rounded down to the nearest integer)

$$prescaler = (ln(SYSCLK/((multiplier + 1) \times f_{Bit} \times 200)/(ln2)) - 1$$

3. Adjusting the bit time divider depending on system clock, data rate, bit time multiplier and prescaler according to the following equation:

$$divider = SYSCLK/(2^{(prescaler + 1)} \times (multiplier + 1) \times f_{BIT})$$

Table 17.3. Bit Timing Examples Without Auto Bit-Rate

SYSCLK	LIN Baud Rate	Multiplier	Prescaler	Divider
24.5 MHz	19.2 KHz	0	1	319
24.5 MHz	9.6 KHz	1	1	319
24.5 MHz	1 KHz	19	1	306

17.4.2. Bit Timing Register adjustment of Slave with Automatic Bit-Rate\

The necessary steps to adjust the Bit-Timing registers of a slave with automatic bit-rate detection are:

1. Setting up the prescaler register depending on system clock according to the following equation:

$$prescaler = (ln(SYSCLK/(20\ KHz \times 200))/ln2) - 1$$

2. Adjusting the bit time divider depending on system clock and prescaler according to the following equation:

$$divider = SYSCLK/(2^{(prescaler + 1)} \times 20\ KHz)$$

Table 17.4. Bit Timing Examples With Auto Bit-Rate

SYSCLK	Prescaler	Divider
8 MHz	0	200
12, 25 MHz	0	306
24.5 MHz	1	306

17.4.3. LIN Master Mode Operation

The operation setup of the LIN peripheral in Master mode requires that the LIN bus is not active (**ACTIVE** bit in **LINST** register set to '0').

The master is responsible for the schedule of the messages. It sends the header of each frame that contains SYNC BREAK FIELD, SYNC FIELD and IDENTIFIER FIELD. The steps for scheduling a message frame are explained in the following paragraphs.

1. Load the 6-bit Identifier into the ID register (**LINID**).

2. Load the "data length" in the **LINSIZE** register (number of data bytes or value "1111b" if the data length should be decoded from the identifier) and set the checksum type (classic or enhanced, defined by the **ENHCHK** bit also in the **LINSIZE** register).

3. Adjust the TXRX bit (LINCTRL.5):

"1" - If the current frame is a transmit operation for the master.

"0" – If the current frame is a receive operation for the master.

4. Load the data bytes to transmit into the data buffer (LINDT1 to LINDT8 and transmit operation only).

5. The **STREQ** bit (**LINCTRL**) is set to start the message transfer. After that the LIN peripheral schedules the message frame and request an interrupt if the message transfer is successfully completed or if an error is occurred.

6. The following steps have to be performed by the application when an interrupt is requested.

6a. Check the **DONE** bit and the **ERROR** bit (**LINST**)

6b. Load the received data from the data buffer if the transfer was successful (for receive operation only).

6c. If the transfer was not successful, check the error register to determine the kind of error. Further error handling has to be done by the application.

6d. Set the **RSTINT** and **RSTERR** bits in the status register (**LINST**) to reset the interrupt request and the error flags.

17.4.4. LIN Slave Mode Operation

Once the Baud rate has been selected and the checksum type selected (classic or enhanced) the LIN peripheral can start receiving messages.

Access from application to data buffer and ID registers of the LIN core slave is possible when a data request is pending (**DTREQ** bit in **LINST** register is '1') and also when the LIN bus is not active (**ACTIVE** bit in **LINST** register set to '0').

The LIN peripheral in slave mode detects the header of the message frame sent by the LIN master. If slave synchronization is enabled (autobaud), the slave synchronizes its internal bit time to the master bit time. An interrupt is requested after the reception of the IDENTIFIER FIELD, when an error is detected or when the message transfer is completed.

The following steps have to be done by the application when an interrupt is requested.

1. Check the **DTREQ** bit in the status register (**LINST**) is set.(set when the IDENTIFIER FIELD has been received). If set then:
2. Load the identifier from the **LINID** register and process it
3. Adjust the **TXRX** bit in the control register (**LINCTRL**) (set to "1" if the current frame is a transmit operation for the slave and set to "0" if the current frame is a receive operation for the slave)
4. Load the "data length" in the **LINSIZE** register (number of data bytes or value "1111b" if the data length should be decoded from the identifier)
5. Load the data to transmit into the data buffer. (for transmit operation only)
6. Set the **DTACK** bit in the **LINCTRL** register.

Notes:

1. Steps 1a...1e have to be done during the IN-FRAME RESPONSE SPACE, if the current frame is a transmit operation for the slave; otherwise a timeout will be detected by the master. If the current frame is a receive operation for the slave, steps 1a...1e have to be finished until the reception of the first byte after the IDENTIFIER FIELD. Otherwise, the internal receive buffer of the LIN peripheral will be overwritten and a timeout error will be detected in the LIN peripheral.
2. If the application detects an unknown identifier (e.g. extended identifier) it has to write a '1' to the **STOP** bit (**LINCTRL**) instead of setting the **DTACK** bit (steps 1b...1e can then be skipped). In that case the LIN peripheral stops the processing of the LIN communication until the next SYNC BREAK is received.
3. Changing the setup of the checksum (classic to enhanced or vice versa) during a transaction will cause the interface to reset and the transaction to be lost. Therefore no change in the checksum should be performed while there is a transaction in progress.
4. The same applies to changes in the LIN interface mode from slave mode to master mode and from master mode to slave mode.
5. Check the **DONE** bit in the status register if the **DTREQ** bit is not set. The transmission was successful if the **DONE** bit is set.
6. If the transmission was successful and the current frame was a receive operation for the slave, load the received data bytes from the data buffer. Else check the error register (**LINERR**) to determine the kind of error. Further error handling has to be done by the application.
7. Set the **RSTINT** and **RSTERR** bits in the status register (**LINCTRL**) to reset the interrupt request and the error flags.

17.4.5. Sleep Mode and Wake-Up

To reduce the systems power consumption the LIN Protocol Specification defines a Sleep Mode. The message used to broadcast a Sleep Mode request has to be started by the LIN master application in the same way as a normal transmit message. The LIN slave application has to decode the Sleep Mode Frame from Identifier and data bytes. After that, it has to put the LIN slave node into the Sleep Mode by setting the **SLEEP** bit in the control register (**LINCTRL**).

If the **SLEEP** bit in the control register (**LINCTRL**) of the LIN slave application is not set and there is no bus activity for 4 s (specified bus idle timeout) the **IDLTOUT** bit in the status register (**LINST**) is set and an interrupt request is generated. After that the application may assume that the LIN bus is in Sleep Mode and set the **SLEEP** bit in the control register (**LINCTRL**).

Sending a Wakeup signal from the master or any slave node terminates the Sleep Mode of the LIN bus. To send a Wakeup signal, the application has to set the **WUPREQ** bit in the status register (**LINST**). After successful transmission of the wakeup signal the **DONE** bit in the status register (**LINST**) of the master node is set and an interrupt request is generated. The LIN slave does not generate an interrupt request after successful transmission of the Wakeup signal but it generates an interrupt request if the master does not respond to the Wakeup signal within 150 ms. In that case the **ERROR** bit in status register (**LINST**) and **TOUT** bit in **LINERR** register are set. The application has to decide whether to transmit another Wakeup signal or not.

All LIN nodes that detect a wakeup signal will set the **WAKEUP** bit in status register (**LINST**) and generate an interrupt request. After that, the application has to reset the **SLEEP** bit in the control register of the LIN slave.

17.4.6. Error Detection and Handling

The LIN peripheral generates an interrupt request and stops the processing of the current frame if it detects an error. The application has to check the type of error by processing the error register (**LINERR**). After that, it has to reset the error register (**LINERR**) and the **ERROR** bit in status register (**LINST**) by writing a ‘1’ to the **RSTERR** bit in the control register (**LINCTRL**). Starting a new message with the LIN peripheral selected as master or sending a Wakeup signal with the LIN peripheral selected as a master or slave is possible only if **ERROR** bit in status register is set to ‘0’.

NOTES:

18. Enhanced Serial Peripheral Interface (SPI0)

The Serial Peripheral Interface (SPI0) provides access to a flexible, full-duplex synchronous serial bus. SPI0 can operate as a master or slave device in both 3-wire or 4-wire modes, and supports multiple masters and slaves on a single SPI bus. The slave-select (NSS) signal can be configured as an input to select SPI0 in slave mode, or to disable Master Mode operation in a multi-master environment, avoiding contention on the SPI bus when more than one master attempts simultaneous data transfers. NSS can also be configured as a chip-select output in master mode, or disabled for 3-wire operation. Additional general purpose port I/O pins can be used to select multiple slave devices in master mode.

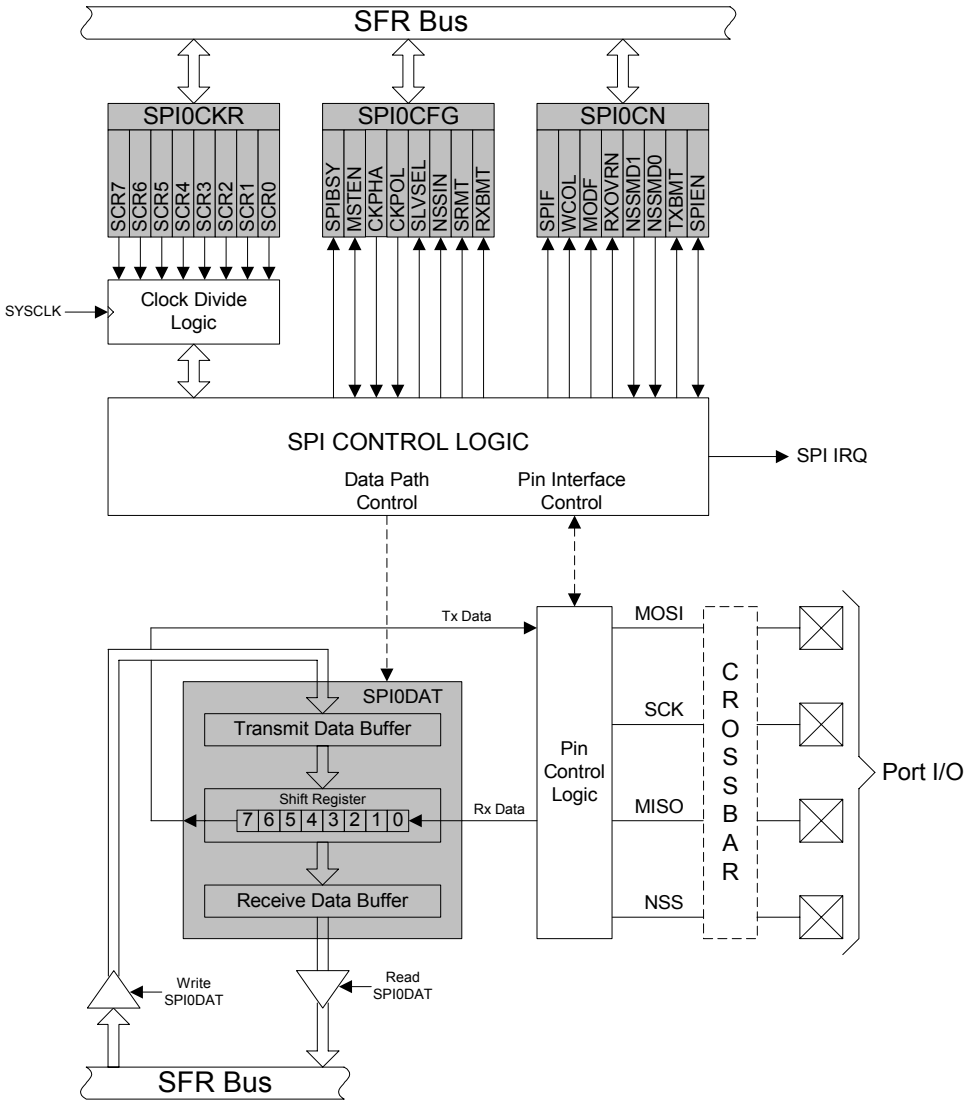


Figure 18.1. SPI Block Diagram

18.1. Signal Descriptions

The four signals used by SPI0 (MOSI, MISO, SCK, NSS) are described below.

18.1.1. Master Out, Slave In (MOSI)

The master-out, slave-in (MOSI) signal is an output from a master device and an input to slave devices. It is used to serially transfer data from the master to the slave. This signal is an output when SPI0 is operating as a master and an input when SPI0 is operating as a slave. Data is transferred most-significant bit first. When configured as a master, MOSI is driven by the MSB of the shift register in both 3- and 4-wire mode.

18.1.2. Master In, Slave Out (MISO)

The master-in, slave-out (MISO) signal is an output from a slave device and an input to the master device. It is used to serially transfer data from the slave to the master. This signal is an input when SPI0 is operating as a master and an output when SPI0 is operating as a slave. Data is transferred most-significant bit first. The MISO pin is placed in a high-impedance state when the SPI module is disabled and when the SPI operates in 4-wire mode as a slave that is not selected. When acting as a slave in 3-wire mode, MISO is always driven by the MSB of the shift register.

18.1.3. Serial Clock (SCK)

The serial clock (SCK) signal is an output from the master device and an input to slave devices. It is used to synchronize the transfer of data between the master and slave on the MOSI and MISO lines. SPI0 generates this signal when operating as a master. The SCK signal is ignored by a SPI slave when the slave is not selected (NSS = 1) in 4-wire slave mode.

18.1.4. Slave Select (NSS)

The function of the slave-select (NSS) signal is dependent on the setting of the NSSMD1 and NSSMD0 bits in the SPI0CN register. There are three possible modes that can be selected with these bits:

1. NSSMD[1:0] = 00: 3-Wire Master or 3-Wire Slave Mode: SPI0 operates in 3-wire mode, and NSS is disabled. When operating as a slave device, SPI0 is always selected in 3-wire mode. Since no select signal is present, SPI0 must be the only slave on the bus in 3-wire mode. This is intended for point-to-point communication between a master and one slave.
2. NSSMD[1:0] = 01: 4-Wire Slave or Multi-Master Mode: SPI0 operates in 4-wire mode, and NSS is enabled as an input. When operating as a slave, NSS selects the SPI0 device. When operating as a master, a 1-to-0 transition of the NSS signal disables the master function of SPI0 so that multiple master devices can be used on the same SPI bus.
3. NSSMD[1:0] = 1x: 4-Wire Master Mode: SPI0 operates in 4-wire mode, and NSS is enabled as an output. The setting of NSSMD0 determines what logic level the NSS pin will output. This configuration should only be used when operating SPI0 as a master device.

See Figure 18.2, Figure 18.3, and Figure 18.4 for typical connection diagrams of the various operational modes. **Note that the setting of NSSMD bits affects the pinout of the device.** When in 3-wire master or 3-wire slave mode, the NSS pin will not be mapped by the crossbar. In all other modes, the NSS signal will be mapped to a pin on the device. See Section "**14. Port Input/Output**" on page **115** for general purpose port I/O and crossbar information.

18.2. SPI0 Master Mode Operation

A SPI master device initiates all data transfers on a SPI bus. SPI0 is placed in master mode by setting the Master Enable flag (MSTEN, SPI0CN.6). Writing a byte of data to the SPI0 data register (SPI0DAT) when in master mode writes to the transmit buffer. If the SPI shift register is empty, the byte in the transmit buffer is moved to the shift register, and a data transfer begins. The SPI0 master immediately shifts out the data serially on the MOSI line while providing the serial clock on SCK. The SPIF (SPI0CN.7) flag is set to logic 1 at the end of the transfer. If interrupts are enabled, an interrupt request is generated when the SPIF flag is set. While the SPI0 master transfers data to a slave on the MOSI line, the addressed SPI slave device simultaneously transfers data to the SPI master on the MISO line in a full-duplex operation. Therefore, the SPIF flag serves as both a transmit-complete and receive-data-ready flag. The data byte received from the slave is transferred MSB-first into the master's shift register. When a byte is fully shifted into the register, it is moved to the receive buffer where it can be read by the processor by reading SPI0DAT.

When configured as a master, SPI0 can operate in one of three different modes: multi-master mode, 3-wire single-master mode, and 4-wire single-master mode. The default, multi-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In this mode, NSS is an input to the device, and is used to disable the master SPI0 when another master is accessing the bus. When NSS is pulled low in this mode, MSTEN (SPI0CN.6) and SPIEN (SPI0CN.0) are set to 0 to disable the SPI master device, and a Mode Fault is generated (MODF, SPI0CN.5 = 1). Mode Fault will generate an interrupt if enabled. SPI0 must be manually re-enabled in software under these circumstances. In multi-master systems, devices will typically default to being slave devices while they are not acting as the system master device. In multi-master mode, slave devices can be addressed individually (if needed) using general-purpose I/O pins. Figure 18.2 shows a connection diagram between two master devices in multiple-master mode.

3-wire single-master mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. In this mode, NSS is not used and is not mapped to an external port pin through the crossbar. Any slave devices that must be addressed in this mode should be selected using general-purpose I/O pins. Figure 18.3 shows a connection diagram between a master device in 3-wire master mode and a slave device.

4-wire single-master mode is active when NSSMD1 (SPI0CN.3) = 1. In this mode, NSS is configured as an output pin and can be used as a slave-select signal for a single SPI device. In this mode, the output value of NSS is controlled (in software) with the bit NSSMD0 (SPI0CN.2). Additional slave devices can be addressed using general-purpose I/O pins. Figure 18.4 shows a connection diagram for a master device in 4-wire master mode and two slave devices.

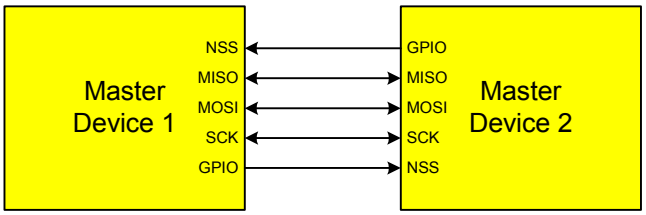


Figure 18.2. Multiple-Master Mode Connection Diagram

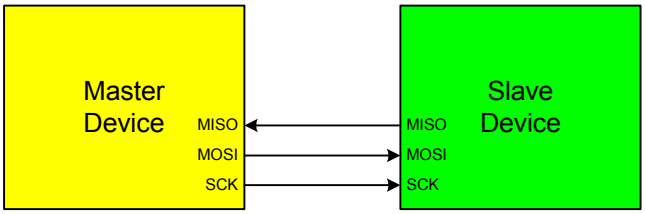


Figure 18.3. 3-Wire Single Master and Slave Mode Connection Diagram

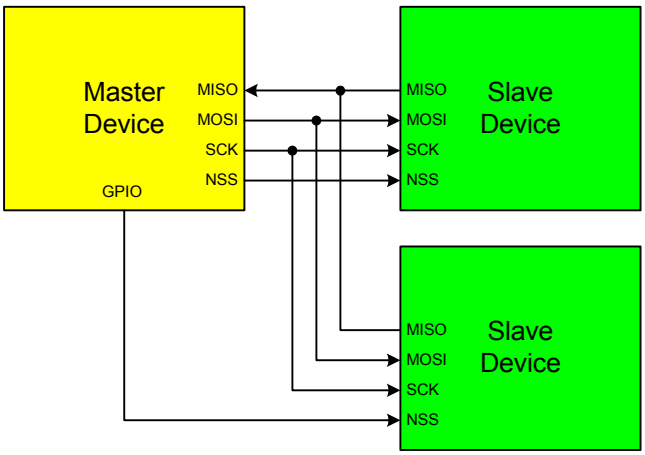


Figure 18.4. 4-Wire Single Master and Slave Mode Connection Diagram

18.3. SPI0 Slave Mode Operation

When SPI0 is enabled and not configured as a master, it will operate as a SPI slave. As a slave, bytes are shifted in through the MOSI pin and out through the MISO pin by a master device controlling the SCK signal. A bit counter in the SPI0 logic counts SCK edges. When 8 bits have been shifted into the shift register, the SPIF flag is set to logic 1, and the byte is copied into the receive buffer. Data is read from the receive buffer by reading SPI0DAT. A slave device cannot initiate transfers. Data to be transferred to the master device is pre-loaded into the shift register by writing to SPI0DAT. Writes to SPI0DAT are double-buffered, and are placed in the transmit buffer first. If the shift register is empty, the contents of the transmit buffer will immediately be transferred into the shift register. When the shift register already contains data, the SPI will load the shift register with the transmit buffer's contents after the last SCK edge of the next (or current) SPI transfer.

The shift register contents are locked after the slave detects the first edge of SCK. Writes to SPI0DAT that occur after the first SCK edge will be held in the TX latch until the end of the current transfer.

When configured as a slave, SPI0 can be configured for 4-wire or 3-wire operation. The default, 4-wire slave mode, is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 1. In 4-wire mode, the NSS signal is routed to a port pin and configured as a digital input. SPI0 is enabled when NSS is logic 0, and disabled when NSS is logic 1. The bit counter is reset on a falling edge of NSS. Note that the NSS signal must be driven low at least 2 system clocks before the first active edge of SCK for each byte transfer. Figure 18.4 shows a connection diagram between two slave devices in 4-wire slave mode and a master device.

3-wire slave mode is active when NSSMD1 (SPI0CN.3) = 0 and NSSMD0 (SPI0CN.2) = 0. NSS is not used in this mode, and is not mapped to an external port pin through the crossbar. Since there is not a way of uniquely addressing the device in 3-wire slave mode, SPI0 must be the only slave device present on the bus. It is important to note that in 3-wire slave mode there is no external means of resetting the bit counter that determines when a full byte has been received. The bit counter can only be reset by disabling and re-enabling SPI0 with the SPIEN bit. Figure 18.3 shows a connection diagram between a slave device in 3-wire slave mode and a master device.

18.4. SPI0 Interrupt Sources

When SPI0 interrupts are enabled, the following four flags will generate an interrupt when they are set to logic 1:

Note that all of the following interrupt bits must be cleared by software.

1. The SPI Interrupt Flag, SPIF (SPI0CN.7) is set to logic 1 at the end of each byte transfer. This flag can occur in all SPI0 modes.
2. The Write Collision Flag, WCOL (SPI0CN.6) is set to logic 1 if a write to SPI0DAT is attempted when the transmit buffer has not been emptied to the SPI shift register. When this occurs, the write to SPI0DAT will be ignored, and the transmit buffer will not be written.This flag can occur in all SPI0 modes.
3. The Mode Fault Flag MODF (SPI0CN.5) is set to logic 1 when SPI0 is configured as a master in multi-master mode and the NSS pin is pulled low. When a Mode Fault occurs, the MSTEN and SPIEN bits in SPI0CN are set to logic 0 to disable SPI0 and allow another master device to access the bus.
4. The Receive Overrun Flag RXOVRN (SPI0CN.4) is set to logic 1 when configured as a slave, and a transfer is completed while the receive buffer still holds an unread byte from a previous transfer. The new byte is not transferred to the receive buffer, allowing the previously received data byte to be read. The data byte which caused the overrun is lost.

18.5. Serial Clock Timing

Four combinations of serial clock phase and polarity can be selected using the clock control bits in the SPI0 Configuration Register (SPI0CFG). The CKPHA bit (SPI0CFG.5) selects one of two clock phases (edge used to latch the data). The CKPOL bit (SPI0CFG.4) selects between a rising edge or a falling edge. Both master and slave devices must be configured to use the same clock phase and polarity. SPI0 should be disabled (by clearing the SPIEN bit, SPI0CN.0) when changing the clock phase or polarity. The clock and data line relationships are shown in Figure 18.5.

The SPI0 Clock Rate Register (SPI0CKR) as shown in SFR Definition 18.3 controls the master mode serial clock frequency. This register is ignored when operating in slave mode. When the SPI is configured as a master, the maximum data transfer rate (bits/sec) is one-half the system clock frequency or 12.5 MHz,

whichever is slower. When the SPI is configured as a slave, the maximum data transfer rate (bits/sec) for full-duplex operation is 1/10 the system clock frequency, provided that the master issues SCK, NSS (in 4-wire slave mode), and the serial input data synchronously with the slave's system clock. If the master issues SCK, NSS, and the serial input data asynchronously, the maximum data transfer rate (bits/sec) must be less than 1/10 the system clock frequency. In the special case where the master only wants to transmit data to the slave and does not need to receive data from the slave (i.e. half-duplex operation), the SPI slave can receive data at a maximum data transfer rate (bits/sec) of 1/4 the system clock frequency. This is provided that the master issues SCK, NSS, and the serial input data synchronously with the slave's system clock.

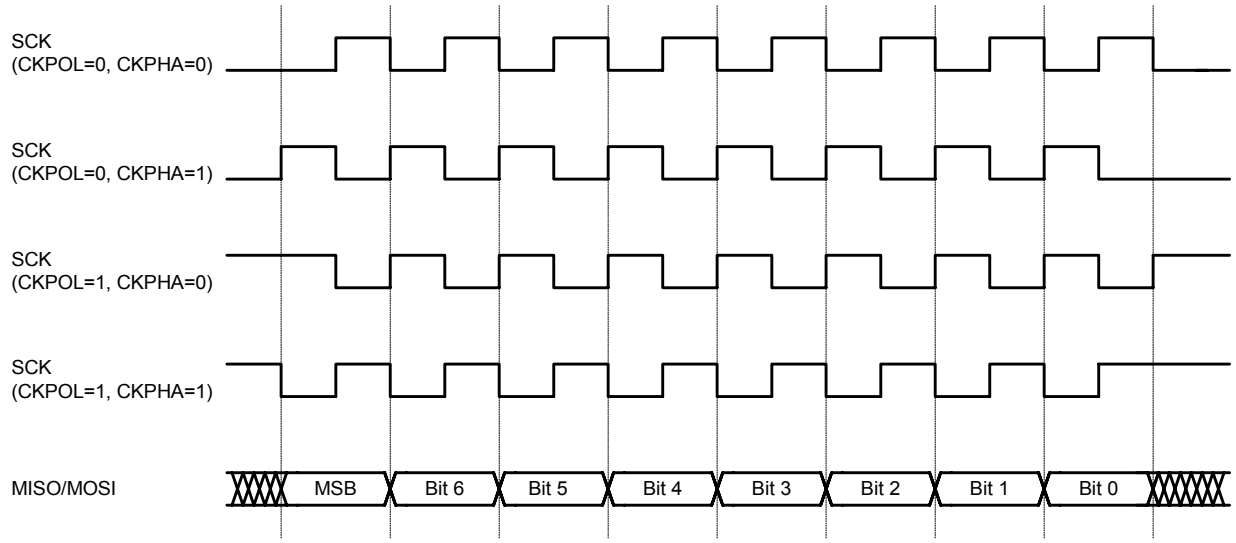


Figure 18.5. Data/Clock Timing Relationship

18.6. SPI Special Function Registers

SPI0 is accessed and controlled through four special function registers in the system controller: SPI0CN Control Register, SPI0DAT Data Register, SPI0CFG Configuration Register, and SPI0CKR Clock Rate Register. The four special function registers related to the operation of the SPI0 Bus are described in the following figures.

SFR Definition 18.1. SPI0CFG: SPI0 Configuration

R	R/W	R/W	R/W	R	R	R	R	Reset Value
SPIBSY	MSTEN	CKPHA	CKPOL	SLVSEL	NSSIN	SRMT	RXBMT	00000111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xA1

Bit 7: **SPIBSY**: SPI Busy (read only).
This bit is set to logic 1 when a SPI transfer is in progress (Master or Slave Mode).

Bit 6: MSTEN: Master Mode Enable.
0: Disable master mode. Operate in slave mode.
1: Enable master mode. Operate as a master.

Bit 5: **CKPHA**: SPI0 Clock Phase.
This bit controls the SPI0 clock phase.
0: Data centered on first edge of SCK period.*
1: Data centered on second edge of SCK period.*

Bit 4: **CKPOL**: SPI0 Clock Polarity.
This bit controls the SPI0 clock polarity.
0: SCK line low in idle state.
1: SCK line high in idle state.

Bit 3: **SLVSEL**: Slave Selected Flag (read only).
This bit is set to logic 1 whenever the NSS pin is low indicating SPI0 is the selected slave. It is cleared to logic 0 when NSS is high (slave not selected). This bit does not indicate the instantaneous value at the NSS pin, but rather a de-glitched version of the pin input.

Bit 2: **NSSIN**: NSS Instantaneous Pin Input (read only).
This bit mimics the instantaneous value that is present on the NSS port pin at the time that the register is read. This input is not de-glitched.

Bit 1: **SRMT**: Shift Register Empty (Valid in Slave Mode, read only).
This bit will be set to logic 1 when all data has been transferred in/out of the shift register, and there is no new information available to read from the transmit buffer or write to the receive buffer. It returns to logic 0 when a data byte is transferred to the shift register from the transmit buffer or by a transition on SCK.
NOTE: SRMT = 1 when in Master Mode.

Bit 0: **RXBMT**: Receive Buffer Empty (Valid in Slave Mode, read only).
This bit will be set to logic 1 when the receive buffer has been read and contains no new information. If there is new information available in the receive buffer that has not been read, this bit will return to logic 0.
NOTE: RXBMT = 1 when in Master Mode.

***Note:** See Table 18.1 for timing parameters.

SFR Definition 18.2. SPI0CN: SPI0 Control

R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	Reset Value
SPIF	WCOL	MODF	RXOVRN	NSSMD1	NSSMD0	TXBMT	SPIEN	00000110
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0xF8

Bit7: **SPIF**: SPI0 Interrupt Flag.
This bit is set to logic 1 by hardware at the end of a data transfer. If interrupts are enabled, setting this bit causes the CPU to vector to the SPI0 interrupt service routine. This bit is not automatically cleared by hardware. It must be cleared by software.

Bit6: **WCOL**: Write Collision Flag.
This bit is set to logic 1 by hardware (and generates a SPI0 interrupt) to indicate a write to the SPI0 data register was attempted while a data transfer was in progress. This bit is not automatically cleared by hardware. It must be cleared by software.

Bit5: **MODF**: Mode Fault Flag.
This bit is set to logic 1 by hardware (and generates a SPI0 interrupt) when a master mode collision is detected (NSS is low, MSTEN = 1, and NSSMD[1:0] = 01). This bit is not automatically cleared by hardware. It must be cleared by software.

Bit4: **RXOVRN**: Receive Overrun Flag (Slave Mode only).
This bit is set to logic 1 by hardware (and generates a SPI0 interrupt) when the receive buffer still holds unread data from a previous transfer and the last bit of the current transfer is shifted into the SPI0 shift register. This bit is not automatically cleared by hardware. It must be cleared by software.

Bits3–2: **NSSMD1–NSSMD0**: Slave Select Mode.
Selects between the following NSS operation modes:
(See **Section "18.2. SPI0 Master Mode Operation" on page 167** and **Section "18.3. SPI0 Slave Mode Operation" on page 168**).
00: 3-Wire Slave or 3-wire Master Mode. NSS signal is not routed to a port pin.
01: 4-Wire Slave or Multi-Master Mode (Default). NSS is always an input to the device.
1x: 4-Wire Single-Master Mode. NSS signal is mapped as an output from the device and will assume the value of NSSMD0.

Bit1: **TXBMT**: Transmit Buffer Empty.
This bit will be set to logic 0 when new data has been written to the transmit buffer. When data in the transmit buffer is transferred to the SPI shift register, this bit will be set to logic 1, indicating that it is safe to write a new byte to the transmit buffer.

Bit0: **SPIEN**: SPI0 Enable.
This bit enables/disables the SPI.
0: SPI disabled.
1: SPI enabled.

SFR Definition 18.3. SPI0CKR: SPI0 Clock Rate

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
SCR7	SCR6	SCR5	SCR4	SCR3	SCR2	SCR1	SCR0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xA2

Bits7–0: **SCR7–SCR0**: SPI0 Clock Rate.
These bits determine the frequency of the SCK output when the SPI0 module is configured for master mode operation. The SCK clock frequency is a divided version of the system clock, and is given in the following equation, where *SYSCLK* is the system clock frequency and *SPI0CKR* is the 8-bit value held in the SPI0CKR register.

$$f_{SCK} = \frac{SYSCLK}{2 \times (SPI0CKR + 1)}$$

for 0 <= SPI0CKR <= 255

Example: If SYSCLK = 2 MHz and SPI0CKR = 0x04,

$$f_{SCK} = \frac{2000000}{2 \times (4 + 1)}$$

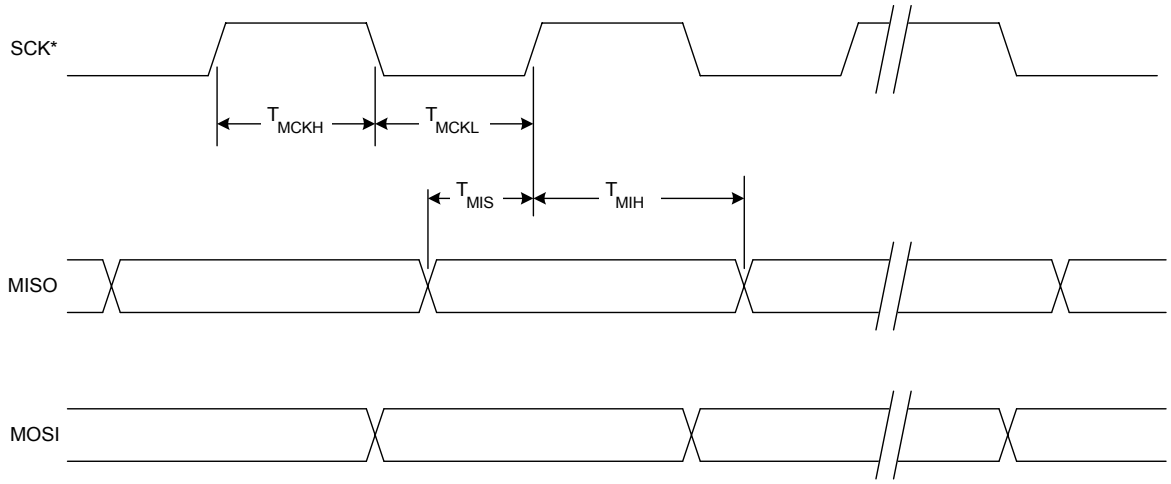
$$f_{SCK} = 200kHz$$

SFR Definition 18.4. SPI0DAT: SPI0 Data

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

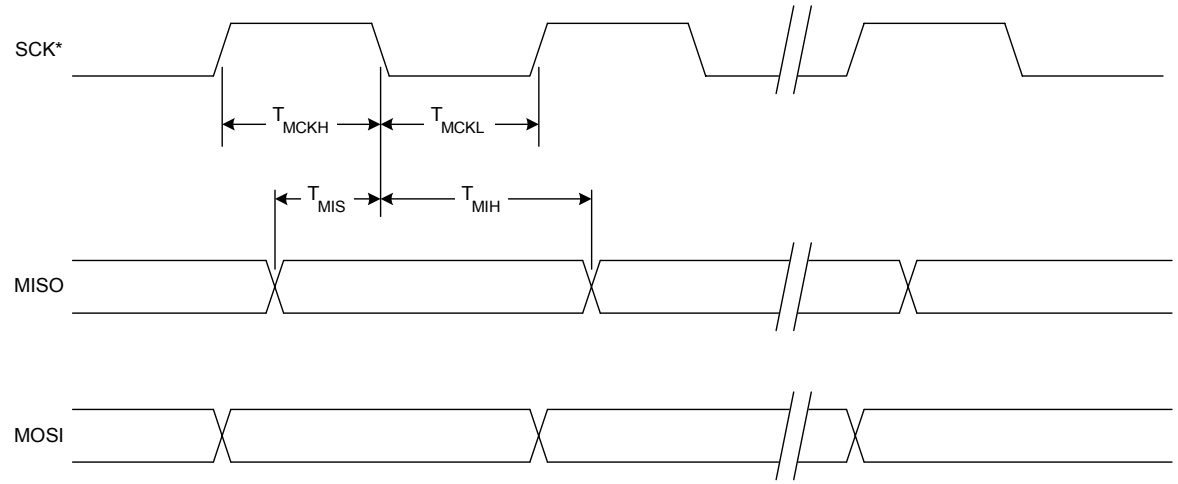
SFR Address: 0xA3

Bits7–0: **SPI0DAT**: SPI0 Transmit and Receive Data.
The SPI0DAT register is used to transmit and receive SPI0 data. Writing data to SPI0DAT places the data into the transmit buffer and initiates a transfer when in Master Mode. A read of SPI0DAT returns the contents of the receive buffer.



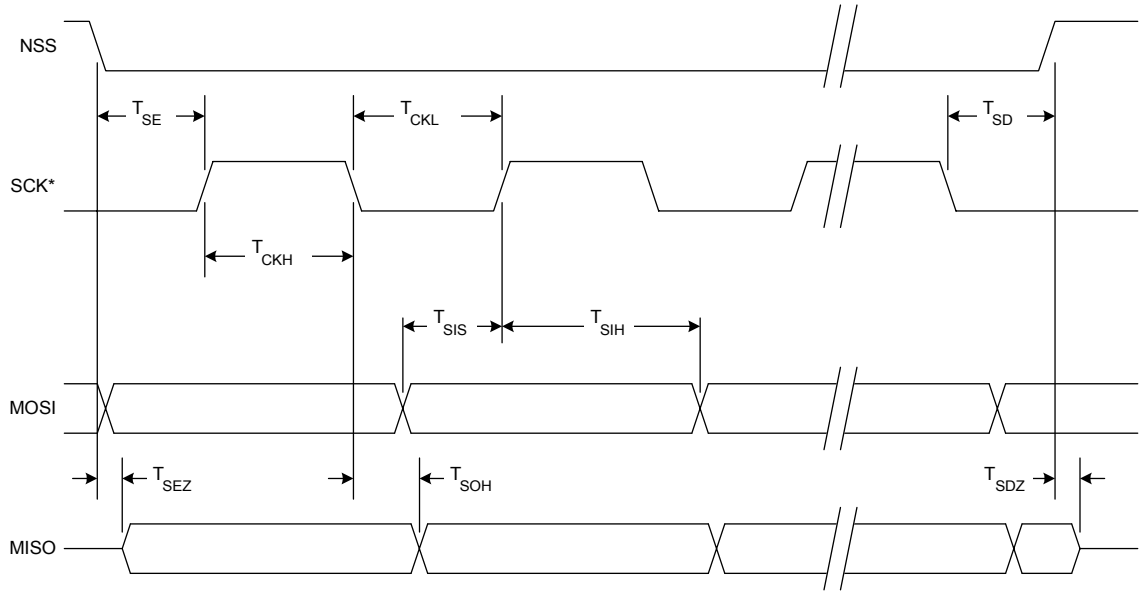
* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 18.6. SPI Master Timing (CKPHA = 0)



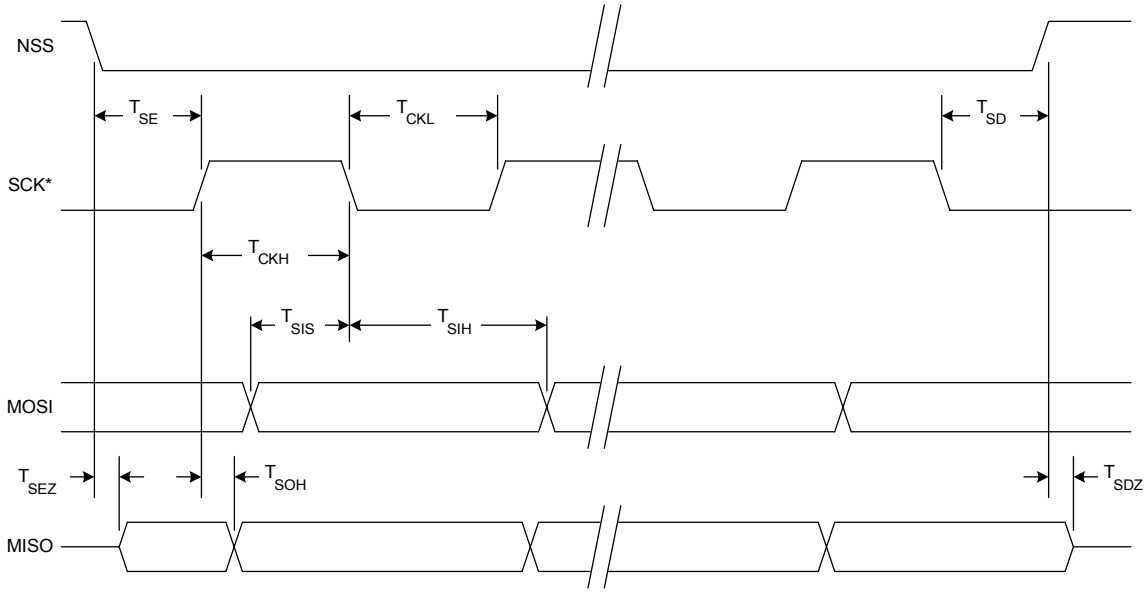
* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 18.7. SPI Master Timing (CKPHA = 1)



* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 18.8. SPI Slave Timing (CKPHA = 0)



* SCK is shown for CKPOL = 0. SCK is the opposite polarity for CKPOL = 1.

Figure 18.9. SPI Slave Timing (CKPHA = 1)

Table 18.1. SPI Slave Timing Parameters

Parameter	Description	Min	Max	Units
Master Mode Timing* (See Figure 18.6 and Figure 18.7)				
T_{MCKH}	SCK High Time	1 x T_{SYSCLK}	—	ns
T_{MCKL}	SCK Low Time	1 x T_{SYSCLK}	—	ns
T_{MIS}	MISO Valid to SCK Sample Edge	20	—	ns
T_{MIH}	SCK Sample Edge to MISO Change	0	—	ns
Slave Mode Timing* (See Figure 18.8 and Figure 18.9)				
T_{SE}	NSS Falling to First SCK Edge	2 x T_{SYSCLK}	—	ns
T_{SD}	Last SCK Edge to NSS Rising	2 x T_{SYSCLK}	—	ns
T_{SEZ}	NSS Falling to MISO Valid	—	4 x T_{SYSCLK}	ns
T_{SDZ}	NSS Rising to MISO High-Z	—	4 x T_{SYSCLK}	ns
T_{CKH}	SCK High Time	5 x T_{SYSCLK}	—	ns
T_{CKL}	SCK Low Time	5 x T_{SYSCLK}	—	ns
T_{SIS}	MOSI Valid to SCK Sample Edge	2 x T_{SYSCLK}	—	ns
T_{SIH}	SCK Sample Edge to MOSI Change	2 x T_{SYSCLK}	—	ns
T_{SOH}	SCK Shift Edge to MISO Change	—	4 x T_{SYSCLK}	ns

***Note:** T_{SYSCLK} is equal to one period of the device system clock (SYSCLK) in ns.
The maximum possible frequency of the SPI can be calculated as:
Transmission: SYSCLK/2
Reception: SYSCLK/10

NOTES:

19. Timers

Each MCU includes three counter/timers: two are 16-bit counter/timers compatible with those found in the standard 8051, and one is a 16-bit auto-reload timer for use with other device peripherals or for general purpose use. These timers can be used to measure time intervals, count external events and generate periodic interrupt requests. Timer 0 and Timer 1 are nearly identical and have four primary modes of operation. Timer 2 offer 16-bit and split 8-bit timer functionality with auto-reload.

Timer 0 and Timer 1 Modes:	Timer 2 Modes:
13-bit counter/timer	16-bit timer with auto-reload
16-bit counter/timer	
8-bit counter/timer with auto-reload	Two 8-bit timers with auto-reload
Two 8-bit counter/timers (Timer 0 only)	

Timers 0 and 1 may be clocked by one of five sources, determined by the Timer Mode Select bits (T1M–T0M) and the Clock Scale bits (SCA1–SCA0). The Clock Scale bits define a pre-scaled clock from which Timer 0 and/or Timer 1 may be clocked (See SFR Definition 19.3 for pre-scaled clock selection).

Timer 0/1 may then be configured to use this pre-scaled clock signal or the system clock. Timer 2 may be clocked by the system clock, the system clock divided by 12, or the external oscillator clock source divided by 8.

Timer 0 and Timer 1 may also be operated as counters. When functioning as a counter, a counter/timer register is incremented on each high-to-low transition at the selected input pin (T0 or T1). Events with a frequency of up to one-fourth the system clock's frequency can be counted. The input signal need not be periodic, but it must be held at a given level for at least two full system clock cycles to ensure the level is properly sampled.

19.1. Timer 0 and Timer 1

Each timer is implemented as a 16-bit register accessed as two separate bytes: a low byte (TL0 or TL1) and a high byte (TH0 or TH1). The Counter/Timer Control register (TCON) is used to enable Timer 0 and Timer 1 as well as indicate status. Timer 0 interrupts can be enabled by setting the ET0 bit in the IE register (**Section “11.4. Interrupt Register Descriptions” on page 91**); Timer 1 interrupts can be enabled by setting the ET1 bit in the IE register (**Section 11.4**). Both counter/timers operate in one of four primary modes selected by setting the Mode Select bits T1M1–T0M0 in the Counter/Timer Mode register (TMOD). Each timer can be configured independently. Each operating mode is described below.

19.1.1. Mode 0: 13-bit Counter/Timer

Timer 0 and Timer 1 operate as 13-bit counter/timers in Mode 0. The following describes the configuration and operation of Timer 0. However, both timers operate identically, and Timer 1 is configured in the same manner as described for Timer 0.

The TH0 register holds the eight MSBs of the 13-bit counter/timer. TL0 holds the five LSBs in bit positions TL0.4–TL0.0. The three upper bits of TL0 (TL0.7–TL0.5) are indeterminate and should be masked out or ignored when reading. As the 13-bit timer register increments and overflows from 0x1FFF (all ones) to 0x0000, the timer overflow flag TF0 (TCON.5) is set and an interrupt will occur if Timer 0 interrupts are enabled.

The C/T0 bit (TMOD.2) selects the counter/timer's clock source. When C/T0 is set to logic 1, high-to-low transitions at the selected Timer 0 input pin (T0) increment the timer register (Refer to **Section "14.1. Priority Crossbar Decoder" on page 117** for information on selecting and configuring external I/O pins). Clearing C/T selects the clock defined by the T0M bit (CKCON.3). When T0M is set, Timer 0 is clocked by the system clock. When T0M is cleared, Timer 0 is clocked by the source selected by the Clock Scale bits in CKCON (see SFR Definition 19.3).

Setting the TR0 bit (TCON.4) enables the timer when either GATE0 (TMOD.3) is logic 0 or the input signal /INT0 is active as defined by bit IN0PL in register IT01CF (see SFR Definition 11.5. IT01CF: INT0/INT1 Configuration). Setting GATE0 to '1' allows the timer to be controlled by the external input signal /INT0 (see **Section "11.4. Interrupt Register Descriptions" on page 91**), facilitating pulse width measurements.

TR0	GATE0	/INT0	Counter/Timer
0	X	X	Disabled
1	0	X	Enabled
1	1	0	Disabled
1	1	1	Enabled
X = Don't Care			

Setting TR0 does not force the timer to reset. The timer registers should be loaded with the desired initial value before the timer is enabled.

TL1 and TH1 form the 13-bit register for Timer 1 in the same manner as described above for TL0 and TH0. Timer 1 is configured and controlled using the relevant TCON and TMOD bits just as with Timer 0. The input signal /INT1 is used with Timer 1; the /INT1 polarity is defined by bit IN1PL in register IT01CF (see SFR Definition 11.5. IT01CF: INT0/INT1 Configuration).

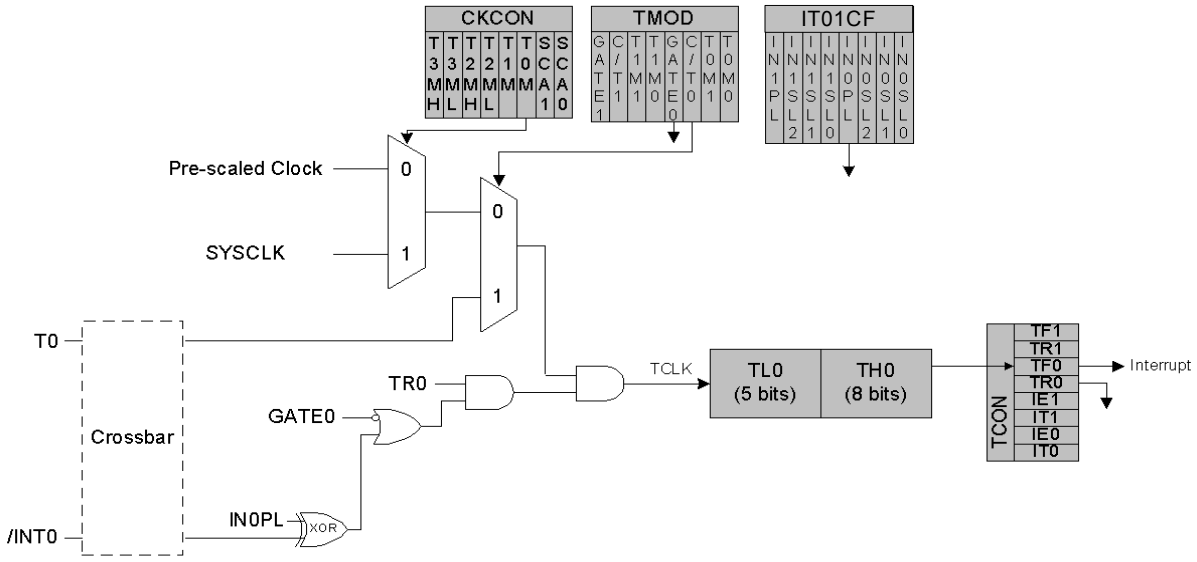


Figure 19.1. T0 Mode 0 Block Diagram

19.1.2. Mode 1: 16-bit Counter/Timer

Mode 1 operation is the same as Mode 0, except that the counter/timer registers use all 16 bits. The counter/timers are enabled and configured in Mode 1 in the same manner as for Mode 0.

19.1.3. Mode 2: 8-bit Counter/Timer with Auto-Reload

Mode 2 configures Timer 0 and Timer 1 to operate as 8-bit counter/timers with automatic reload of the start value. TL0 holds the count and TH0 holds the reload value. When the counter in TL0 overflows from all ones to 0x00, the timer overflow flag TF0 (TCON.5) is set and the counter in TL0 is reloaded from TH0. If Timer 0 interrupts are enabled, an interrupt will occur when the TF0 flag is set. The reload value in TH0 is not changed. TL0 must be initialized to the desired value before enabling the timer for the first count to be correct. When in Mode 2, Timer 1 operates identically to Timer 0.

Both counter/timers are enabled and configured in Mode 2 in the same manner as Mode 0. Setting the TR0 bit (TCON.4) enables the timer when either GATE0 (TMOD.3) is logic 0 or when the input signal /INT0 is active as defined by bit IN0PL in register IT01CF (see **Section "11.5. External Interrupts" on page 95** for details on the external input signals /INT0 and /INT1).

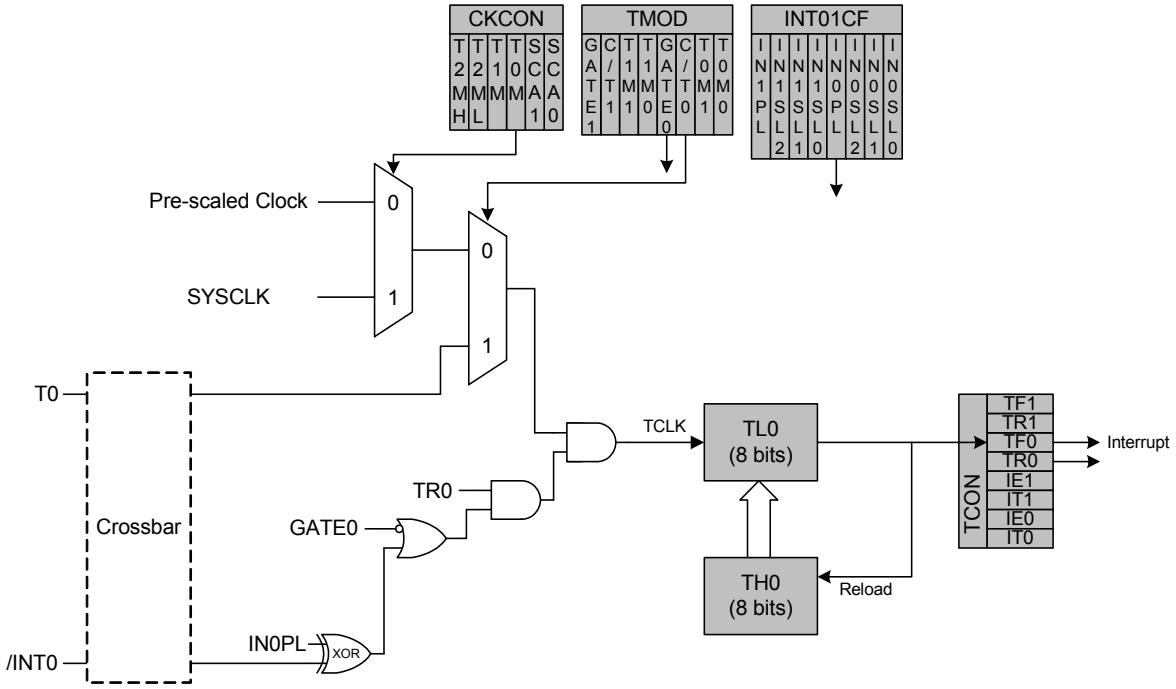


Figure 19.2. T0 Mode 2 Block Diagram

19.1.4. Mode 3: Two 8-bit Counter/Timers (Timer 0 Only)

In Mode 3, Timer 0 is configured as two separate 8-bit counter/timers held in TL0 and TH0. The counter/timer in TL0 is controlled using the Timer 0 control/status bits in TCON and TMOD: TR0, C/T0, GATE0 and TF0. TL0 can use either the system clock or an external input signal as its timebase. The TH0 register is restricted to a timer function sourced by the system clock or prescaled clock. TH0 is enabled using the Timer 1 run control bit TR1. TH0 sets the Timer 1 overflow flag TF1 on overflow and thus controls the Timer 1 interrupt.

Timer 1 is inactive in Mode 3. When Timer 0 is operating in Mode 3, Timer 1 can be operated in Modes 0, 1 or 2, but cannot be clocked by external signals nor set the TF1 flag and generate an interrupt. However, the Timer 1 overflow can be used to generate baud rates for the SMBus and UART. While Timer 0 is operating in Mode 3, Timer 1 run control is handled through its mode settings. To run Timer 1 while Timer 0 is in Mode 3, set the Timer 1 Mode as 0, 1, or 2. To disable Timer 1, configure it for Mode 3.

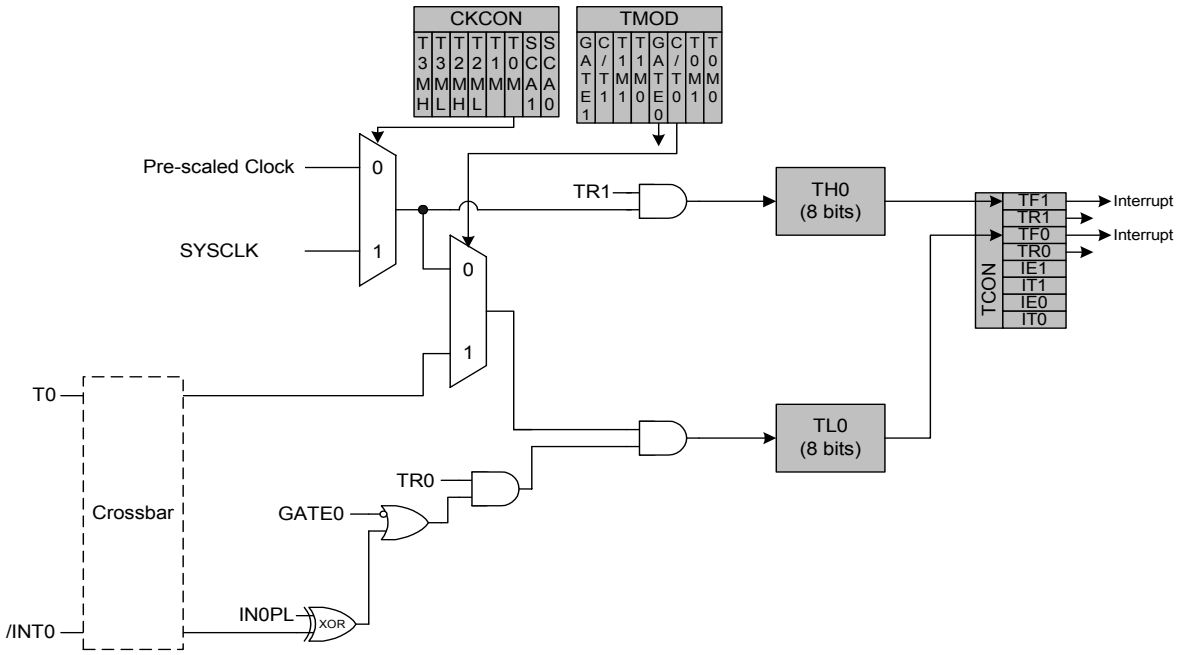


Figure 19.3. T0 Mode 3 Block Diagram

SFR Definition 19.1. TCON: Timer Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0x88

Bit7: **TF1**: Timer 1 Overflow Flag.
Set by hardware when Timer 1 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 1 interrupt service routine.
0: No Timer 1 overflow detected.
1: Timer 1 has overflowed.

Bit6: **TR1**: Timer 1 Run Control.
0: Timer 1 disabled.
1: Timer 1 enabled.

Bit5: **TF0**: Timer 0 Overflow Flag.
Set by hardware when Timer 0 overflows. This flag can be cleared by software but is automatically cleared when the CPU vectors to the Timer 0 interrupt service routine.
0: No Timer 0 overflow detected.
1: Timer 0 has overflowed.

Bit4: **TR0**: Timer 0 Run Control.
0: Timer 0 disabled.
1: Timer 0 enabled.

Bit3: **IE1**: External Interrupt 1.
This flag is set by hardware when an edge/level of type defined by IT1 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 1 service routine if IT1 = 1. When IT1 = 0, this flag is set to ‘1’ when /INT1 is active as defined by bit IN1PL in register IT01CF (see SFR Definition 11.5. “IT01CF: INT0/INT1 Configuration” on page 96).

Bit2: **IT1**: Interrupt 1 Type Select.
This bit selects whether the configured /INT1 interrupt will be edge or level sensitive. /INT1 is configured active low or high by the IN1PL bit in the IT01CF register (see SFR Definition 11.5. “IT01CF: INT0/INT1 Configuration” on page 96).
0: /INT1 is level triggered.
1: /INT1 is edge triggered.

Bit1: **IE0**: External Interrupt 0.
This flag is set by hardware when an edge/level of type defined by IT0 is detected. It can be cleared by software but is automatically cleared when the CPU vectors to the External Interrupt 0 service routine if IT0 = 1. When IT0 = 0, this flag is set to ‘1’ when /INT0 is active as defined by bit IN0PL in register IT01CF (see SFR Definition 11.5. “IT01CF: INT0/INT1 Configuration” on page 96).

Bit0: **IT0**: Interrupt 0 Type Select.
This bit selects whether the configured /INT0 interrupt will be edge or level sensitive. /INT0 is configured active low or high by the IN0PL bit in register IT01CF (see SFR Definition 11.5. “IT01CF: INT0/INT1 Configuration” on page 96).
0: /INT0 is level triggered.
1: /INT0 is edge triggered.

SFR Definition 19.2. TMOD: Timer Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
GATE1	C/T1	T1M1	T1M0	GATE0	C/T0	T0M1	T0M0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x89

Bit7: **GATE1**: Timer 1 Gate Control.
0: Timer 1 enabled when TR1 = 1 irrespective of /INT1 logic level.
1: Timer 1 enabled only when TR1 = 1 AND /INT1 is active as defined by bit IN1PL in register IT01CF (see SFR Definition 11.5. "IT01CF: INT0/INT1 Configuration" on page 96).

Bit6: **C/T1**: Counter/Timer 1 Select.
0: Timer Function: Timer 1 incremented by clock defined by T1M bit (CKCON.4).
1: Counter Function: Timer 1 incremented by high-to-low transitions on external input pin (T1).

Bits5–4: **T1M1–T1M0**: Timer 1 Mode Select.
These bits select the Timer 1 operation mode.

T1M1	T1M0	Mode
0	0	Mode 0: 13-bit counter/timer
0	1	Mode 1: 16-bit counter/timer
1	0	Mode 2: 8-bit counter/timer with auto-reload
1	1	Mode 3: Timer 1 inactive

Bit3: **GATE0**: Timer 0 Gate Control.
0: Timer 0 enabled when TR0 = 1 irrespective of /INT0 logic level.
1: Timer 0 enabled only when TR0 = 1 AND /INT0 is active as defined by bit IN0PL in register IT01CF (see SFR Definition 11.5. "IT01CF: INT0/INT1 Configuration" on page 96).

Bit2: **C/T0**: Counter/Timer Select.
0: Timer Function: Timer 0 incremented by clock defined by T0M bit (CKCON.3).
1: Counter Function: Timer 0 incremented by high-to-low transitions on external input pin (T0).

Bits1–0: **T0M1–T0M0**: Timer 0 Mode Select.
These bits select the Timer 0 operation mode.

T0M1	T0M0	Mode
0	0	Mode 0: 13-bit counter/timer
0	1	Mode 1: 16-bit counter/timer
1	0	Mode 2: 8-bit counter/timer with auto-reload
1	1	Mode 3: Two 8-bit counter/timers

SFR Definition 19.3. CKCON: Clock Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
—	—	T2MH	T2ML	T1M	T0M	SCA1	SCA0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x8E

Bit7–6: **RESERVED**. Read = 0b; Must write 0b.

Bit5: **T2MH**: Timer 2 High Byte Clock Select.
This bit selects the clock supplied to the Timer 2 high byte if Timer 2 is configured in split 8-bit timer mode. T2MH is ignored if Timer 2 is in any other mode.
0: Timer 2 high byte uses the clock defined by the T2XCLK bit in TMR2CN.
1: Timer 2 high byte uses the system clock.

Bit4: **T2ML**: Timer 2 Low Byte Clock Select.
This bit selects the clock supplied to Timer 2. If Timer 2 is configured in split 8-bit timer mode, this bit selects the clock supplied to the lower 8-bit timer.
0: Timer 2 low byte uses the clock defined by the T2XCLK bit in TMR2CN.
1: Timer 2 low byte uses the system clock.

Bit3: **T1M**: Timer 1 Clock Select.
This select the clock source supplied to Timer 1. T1M is ignored when C/T1 is set to logic 1.
0: Timer 1 uses the clock defined by the prescale bits, SCA1–SCA0.
1: Timer 1 uses the system clock.

Bit2: **T0M**: Timer 0 Clock Select.
This bit selects the clock source supplied to Timer 0. T0M is ignored when C/T0 is set to logic 1.
0: Counter/Timer 0 uses the clock defined by the prescale bits, SCA1–SCA0.
1: Counter/Timer 0 uses the system clock.

Bits1–0: **SCA1–SCA0**: Timer 0/1 Prescale Bits.
These bits control the division of the clock supplied to Timer 0 and Timer 1 if configured to use prescaled clock inputs.

SCA1	SCA0	Prescaled Clock
0	0	System clock divided by 12
0	1	System clock divided by 4
1	0	System clock divided by 48
1	1	External clock divided by 8

Note: External clock divided by 8 is synchronized with the system clock.

SFR Definition 19.4. TL0: Timer 0 Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x8A

Bits 7–0: **TL0**: Timer 0 Low Byte.
The TL0 register is the low byte of the 16-bit Timer 0.

SFR Definition 19.5. TL1: Timer 1 Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x8B

Bits 7–0: **TL1**: Timer 1 Low Byte.
The TL1 register is the low byte of the 16-bit Timer 1.

SFR Definition 19.6. TH0: Timer 0 High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x8C

Bits 7–0: **TH0**: Timer 0 High Byte.
The TH0 register is the high byte of the 16-bit Timer 0.

SFR Definition 19.7. TH1: Timer 1 High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0x8D

Bits 7–0: **TH1**: Timer 1 High Byte.
The TH1 register is the high byte of the 16-bit Timer 1.

19.2. Timer 2

Timer 2 is a 16-bit timer formed by two 8-bit SFRs: TMR2L (low byte) and TMR2H (high byte). Timer 2 may operate in 16-bit auto-reload mode or (split) 8-bit auto-reload mode. The T2SPLIT bit (TMR2CN.3) defines the Timer 2 operation mode. Timer 2 can also be used in Capture Mode to measure the RTC0 clock frequency or the External Oscillator clock frequency.

Timer 2 may be clocked by the system clock, the system clock divided by 12, or the external oscillator source divided by 8. The external oscillator source divided by 8 is synchronized with the system clock.

19.2.1. 16-bit Timer with Auto-Reload

When T2SPLIT (TMR2CN.3) is zero, Timer 2 operates as a 16-bit timer with auto-reload. Timer 2 can be clocked by SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. As the 16-bit timer register increments and overflows from 0xFFFF to 0x0000, the 16-bit value in the Timer 2 reload registers (TMR2RLH and TMR2RLL) is loaded into the Timer 2 register as shown in Figure 19.4, and the Timer 2 High Byte Overflow Flag (TMR2CN.7) is set. If Timer 2 interrupts are enabled (if IE.5 is set), an interrupt will be generated on each Timer 2 overflow. Additionally, if Timer 2 interrupts are enabled and the TF2LEN bit is set (TMR2CN.5), an interrupt will be generated each time the lower 8 bits (TMR2L) overflow from 0xFF to 0x00.

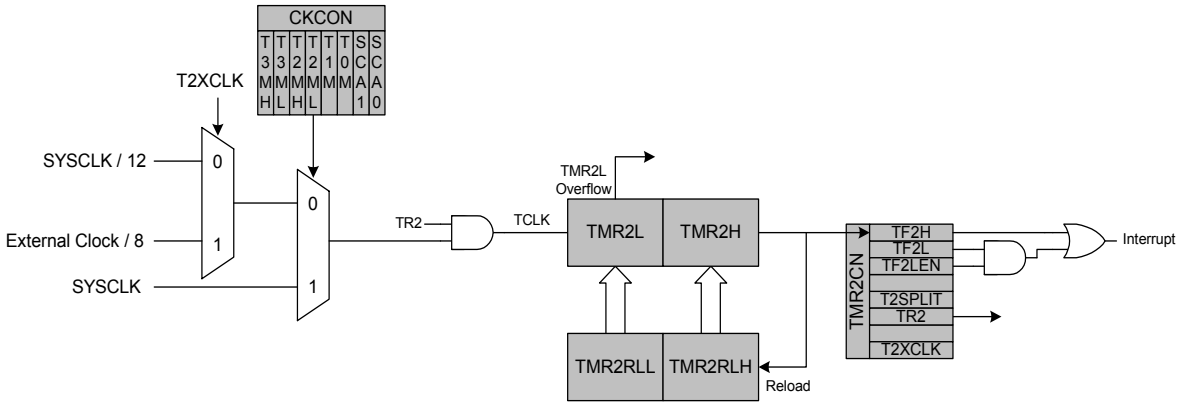


Figure 19.4. Timer 2 16-Bit Mode Block Diagram

19.2.2. 8-bit Timers with Auto-Reload

When T2SPLIT is set, Timer 2 operates as two 8-bit timers (TMR2H and TMR2L). Both 8-bit timers operate in auto-reload mode as shown in Figure 19.5. TMR2RLL holds the reload value for TMR2L; TMR2RLH holds the reload value for TMR2H. The TR2 bit in TMR2CN handles the run control for TMR2H. TMR2L is always running when configured for 8-bit Mode.

Each 8-bit timer may be configured to use SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. The Timer 2 Clock Select bits (T2MH and T2ML in CKCON) select either SYSCLK or the clock defined by the Timer 2 External Clock Select bit (T2XCLK in TMR2CN), as follows:

T2MH	T2XCLK	TMR2H Clock Source
0	0	SYSCLK / 12
0	1	External Clock / 8
1	X	SYSCLK

T2ML	T2XCLK	TMR2L Clock Source
0	0	SYSCLK / 12
0	1	External Clock / 8
1	X	SYSCLK

The TF2H bit is set when TMR2H overflows from 0xFF to 0x00; the TF2L bit is set when TMR2L overflows from 0xFF to 0x00. When Timer 2 interrupts are enabled (IE.5), an interrupt is generated each time TMR2H overflows. If Timer 2 interrupts are enabled and TF2LEN (TMR2CN.5) is set, an interrupt is generated each time either TMR2L or TMR2H overflows. When TF2LEN is enabled, software must check the TF2H and TF2L flags to determine the source of the Timer 2 interrupt. The TF2H and TF2L interrupt flags are not cleared by hardware and must be manually cleared by software.

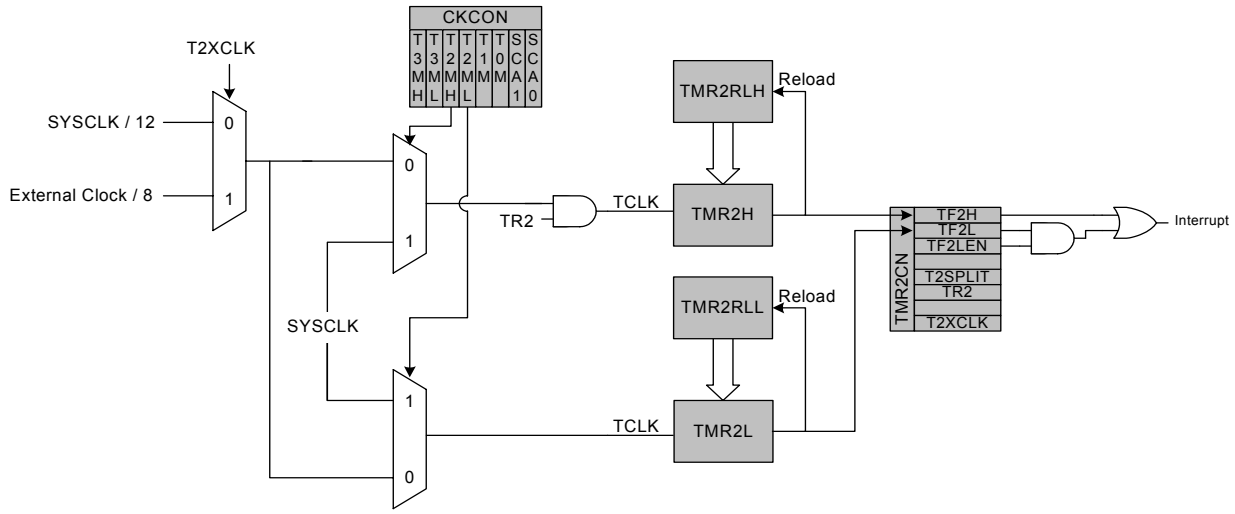


Figure 19.5. Timer 2 8-Bit Mode Block Diagram

19.2.3. External Capture Mode

Capture Mode allows either the external oscillator to be measured against the system clock. The external oscillator clock can also be compared against each other. Timer 2 can be clocked from the system clock, the system clock divided by 12, the external oscillator divided by 8, depending on the T2ML (CKCON.4), T2XCLK, and T2RCLK settings. The timer will capture either every 8 eternal clock cycles, depending on the T2RCLK setting. When a capture event is generated, the contents of Timer 2 (TMR2H:TMR2L) are loaded into the Timer 2 reload registers (TMR2RLH:TMR2RLL) and the TF2H flag is set. By recording the difference between two successive timer capture values, the external oscillator can be determined with respect to the Timer 2 clock. The Timer 2 clock should be much faster than the capture clock to achieve an accurate reading. Timer 2 should be in 16-bit auto-reload mode when using Capture Mode.

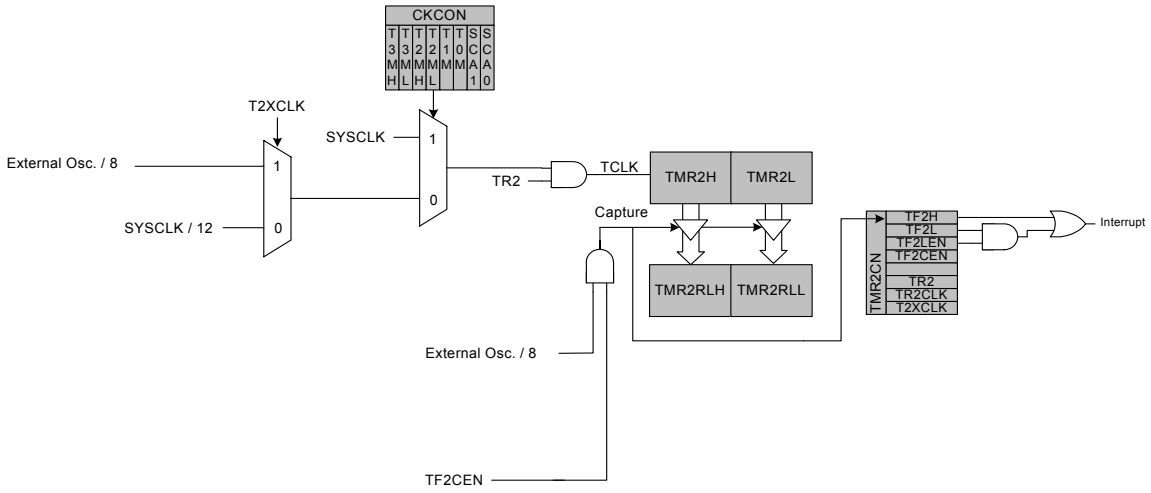


Figure 19.6. Timer 2 Capture Mode Block Diagram

SFR Definition 19.8. TMR2CN: Timer 2 Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
TF2H	TF2L	TF2LEN	TF2CEN	T2SPLIT	TR2	T2RCLK	T2XCLK	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0xC8

Bit7: **TF2H**: Timer 2 High Byte Overflow Flag.
Set by hardware when the Timer 2 high byte overflows from 0xFF to 0x00. In 16 bit mode, this will occur when Timer 2 overflows from 0xFFFF to 0x0000. When the Timer 2 interrupt is enabled, setting this bit causes the CPU to vector to the Timer 2 interrupt service routine. TF2H is not automatically cleared by hardware and must be cleared by software.

Bit6: **TF2L**: Timer 2 Low Byte Overflow Flag.
Set by hardware when the Timer 2 low byte overflows from 0xFF to 0x00. When this bit is set, an interrupt will be generated if TF2LEN is set and Timer 2 interrupts are enabled. TF2L will set when the low byte overflows regardless of the Timer 2 mode. This bit is not automatically cleared by hardware.

Bit5: **TF2LEN**: Timer 2 Low Byte Interrupt Enable.
This bit enables/disables Timer 2 Low Byte interrupts. If TF2LEN is set and Timer 2 interrupts are enabled, an interrupt will be generated when the low byte of Timer 2 overflows. This bit should be cleared when operating Timer 2 in 16-bit mode.
0: Timer 2 Low Byte interrupts disabled.
1: Timer 2 Low Byte interrupts enabled.

Bit4: **TF2CEN**. Timer 2 Capture Enable.
0: Timer 2 capture mode disabled.
1: Timer 2 capture mode enabled.

Bit3: **T2SPLIT**: Timer 2 Split Mode Enable.
When this bit is set, Timer 2 operates as two 8-bit timers with auto-reload.
0: Timer 2 operates in 16-bit auto-reload mode.
1: Timer 2 operates as two 8-bit auto-reload timers.

Bit2: **TR2**: Timer 2 Run Control.
This bit enables/disables Timer 2. In 8-bit mode, this bit enables/disables TMR2H only; TMR2L is always enabled in this mode.
0: Timer 2 disabled.
1: Timer 2 enabled.

Bit1: **T2RCLK**: Timer 2 Capture Mode.
This bit controls the Timer 2 capture source when TF2CEN=1. If T2XCLK = 1 and T2ML (CKCON.4) = 0, this bit also controls the clock source for Timer 2.
0: Capture every RTC clock/8. If T2XCLK = 1 and T2ML (CKCON.4) = 0, count at external oscillator/8.
1: Capture every external oscillator/8. If T2XCLK = 1 and T2ML (CKCON.4) = 0, count at RTC0 clock/8.

Bit0: **T2XCLK**: Timer 2 External Clock Select.
This bit selects the external clock source for Timer 2. If Timer 2 is in 8-bit mode, this bit selects the external oscillator clock source for both timer bytes. However, the Timer 2 Clock Select bits (T2MH and T2ML in register CKCON) may still be used to select between the external clock and the system clock for either timer.
0: Timer 2 external clock selection is the system clock divided by 12.
1: Timer 2 external clock uses the clock defined by the T2RCLK bit.

SFR Definition 19.9. TMR2RLL: Timer 2 Reload Register Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xCA

Bits7–0: **TMR2RLL**: Timer 2 Reload Register Low Byte.
TMR2RLL holds the low byte of the reload value for Timer 2.

SFR Definition 19.10. TMR2RLH: Timer 2 Reload Register High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xCB

Bits7–0: **TMR2RLH**: Timer 2 Reload Register High Byte.
The TMR2RLH holds the high byte of the reload value for Timer 2.

SFR Definition 19.11. TMR2L: Timer 2 Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xCC

Bits7–0: **TMR2L**: Timer 2 Low Byte.
In 16-bit mode, the TMR2L register contains the low byte of the 16-bit Timer 2. In 8-bit mode, TMR2L contains the 8-bit low byte timer value.

SFR Definition 19.12. TMR2H Timer 2 High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xCD

Bits7–0: **TMR2H**: Timer 2 High Byte.
In 16-bit mode, the TMR2H register contains the high byte of the 16-bit Timer 2. In 8-bit mode, TMR2H contains the 8-bit high byte timer value.

NOTES:

20. Programmable Counter Array (PCA0)

The Programmable Counter Array (PCA0) provides enhanced timer functionality while requiring less CPU intervention than the standard 8051 counter/timers. The PCA consists of a dedicated 16-bit counter/timer and three 16-bit capture/compare modules. Each capture/compare module has its own associated I/O line (CEXn) which is routed through the Crossbar to Port I/O when enabled (See **Section "14.1. Priority Crossbar Decoder" on page 117** for details on configuring the Crossbar). The counter/timer is driven by a programmable timebase that can select between six sources: system clock, system clock divided by four, system clock divided by twelve, the external oscillator clock source divided by 8, Timer 0 overflow, or an external clock signal on the ECI input pin. Each capture/compare module may be configured to operate independently in one of three modes: Edge-Triggered Capture, Software Timer, High-Speed Output, Frequency Output, 8-Bit PWM, or 16-Bit PWM (each mode is described in **Section "20.2. Capture/Compare Modules" on page 195**). The PCA is configured and controlled through the system controller's Special Function Registers. The PCA block diagram is shown in Figure 20.1

Important Note: The PCA Module 2 may be used as a watchdog timer (WDT), and is enabled in this mode following a system reset. **Access to certain PCA registers is restricted while WDT mode is enabled**. See **Section 20.3** for details.

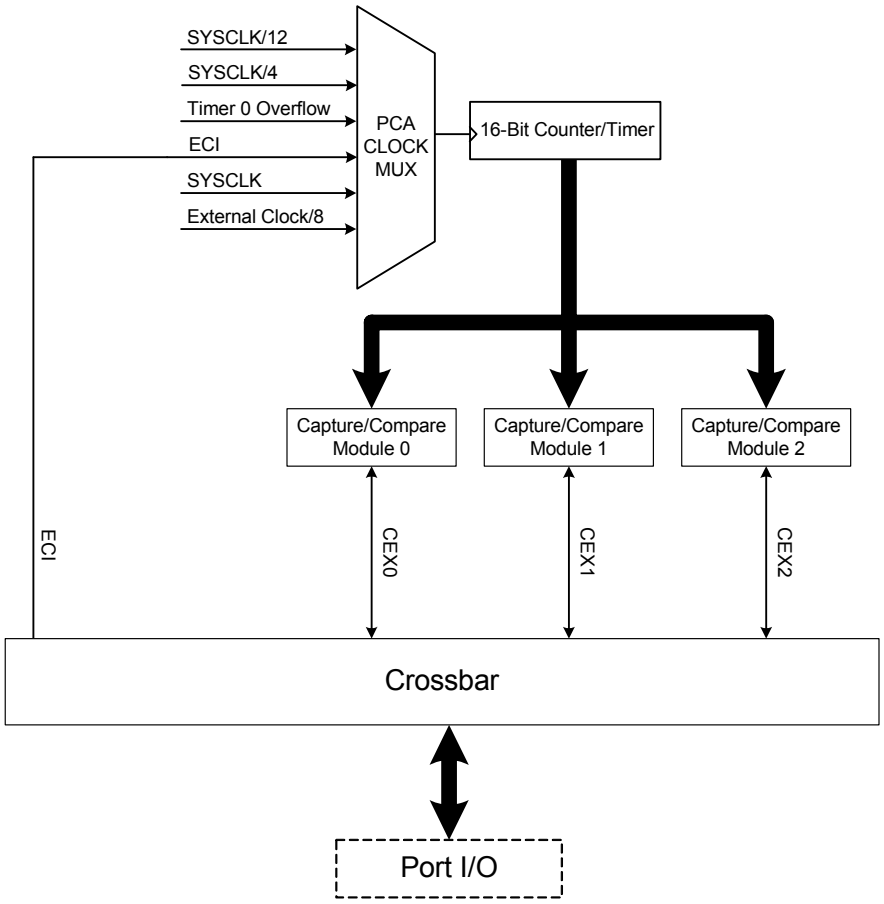


Figure 20.1. PCA Block Diagram

20.1. PCA Counter/Timer

The 16-bit PCA counter/timer consists of two 8-bit SFRs: PCA0L and PCA0H. PCA0H is the high byte (MSB) of the 16-bit counter/timer and PCA0L is the low byte (LSB). Reading PCA0L automatically latches the value of PCA0H into a “snapshot” register; the following PCA0H read accesses this “snapshot” register. **Reading the PCA0L Register first guarantees an accurate reading of the entire 16-bit PCA0 counter.** Reading PCA0H or PCA0L does not disturb the counter operation. The CPS2-CPS0 bits in the PCA0MD register select the timebase for the counter/timer as shown in Table 20.1.

When the counter/timer overflows from 0xFFFF to 0x0000, the Counter Overflow Flag (CF) in PCA0MD is set to logic 1 and an interrupt request is generated if CF interrupts are enabled. Setting the ECF bit in PCA0MD to logic 1 enables the CF flag to generate an interrupt request. The CF bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software (Note: PCA0 interrupts must be globally enabled before CF interrupts are recognized. PCA0 interrupts are globally enabled by setting the EA bit (IE.7) and the EPCA0 bit in EIE1 to logic 1). Clearing the CIDL bit in the PCA0MD register allows the PCA to continue normal operation while the CPU is in Idle mode.

Table 20.1. PCA Timebase Input Options

CPS2	CPS1	CPS0	Timebase
0	0	0	System clock divided by 12
0	0	1	System clock divided by 4
0	1	0	Timer 0 overflow
0	1	1	High-to-low transitions on ECI (max rate = system clock divided by 4)
1	0	0	System clock
1	0	1	External oscillator source divided by 8*
1	1	0	RTC clock divided by 8*

***Note:** External clock divided by 8 and RTC0 clock divided by 8 are synchronized with the system clock.

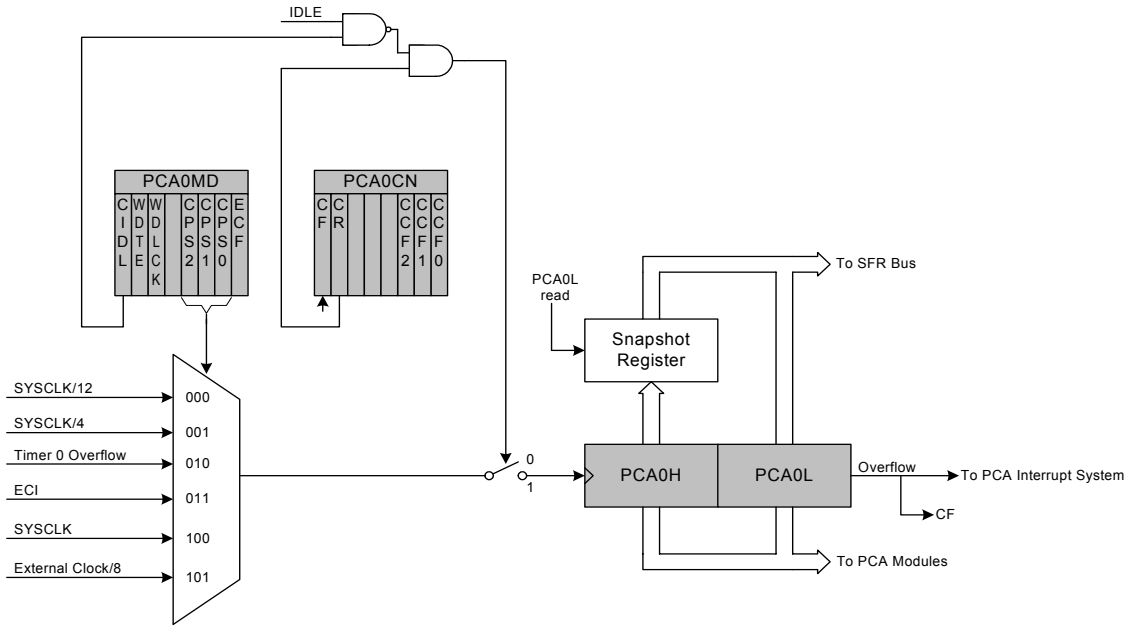


Figure 20.2. PCA Counter/Timer Block Diagram

20.2. Capture/Compare Modules

Each module can be configured to operate independently in one of six operation modes: Edge-triggered Capture, Software Timer, High Speed Output, Frequency Output, 8-Bit Pulse Width Modulator, or 16-Bit Pulse Width Modulator. Each module has Special Function Registers (SFRs) associated with it in the CIP-51 system controller. These registers are used to exchange data with a module and configure the module's mode of operation.

Table 20.2 summarizes the bit settings in the PCA0CPMn registers used to select the PCA capture/compare module's operating modes. Setting the ECCFn bit in a PCA0CPMn register enables the module's CCFn interrupt. Note: PCA0 interrupts must be globally enabled before individual CCFn interrupts are recognized. PCA0 interrupts are globally enabled by setting the EA bit and the EPCA0 bit to logic 1. See Figure 20.3 for details on the PCA interrupt configuration.

Table 20.2. PCA0CPM Register Settings for PCA Capture/Compare Modules

PWM16	ECOM	CAPP	CAPN	MAT	TOG	PWM	ECCF	Operation Mode
X	X	1	0	0	0	0	X	Capture triggered by positive edge on CEXn
X	X	0	1	0	0	0	X	Capture triggered by negative edge on CEXn
X	X	1	1	0	0	0	X	Capture triggered by transition on CEXn
X	1	0	0	1	0	0	X	Software Timer
X	1	0	0	1	1	0	X	High Speed Output
X	1	0	0	X	1	1	X	Frequency Output
0	1	0	0	X	0	1	X	8-Bit Pulse Width Modulator
1	1	0	0	X	0	1	X	16-Bit Pulse Width Modulator
X = Don't Care								

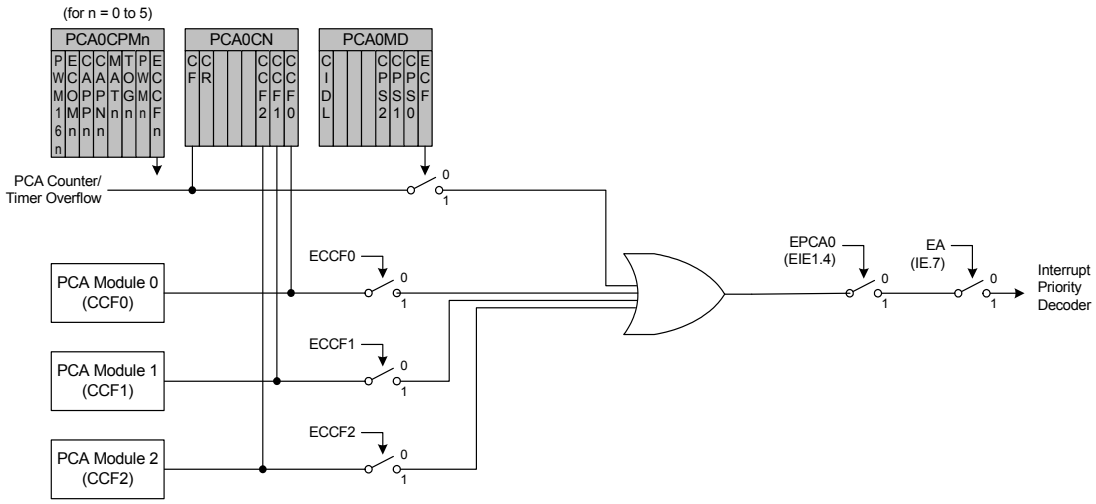


Figure 20.3. PCA Interrupt Block Diagram

20.2.1. Edge-triggered Capture Mode

In this mode, a valid transition on the CEXn pin causes the PCA to capture the value of the PCA counter/timer and load it into the corresponding module's 16-bit capture/compare register (PCA0CPLn and PCA0CPHn). The CAPPn and CAPNn bits in the PCA0CPMn register are used to select the type of transition that triggers the capture: low-to-high transition (positive edge), high-to-low transition (negative edge), or either transition (positive or negative edge). When a capture occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic 1 and an interrupt request is generated if CCF interrupts are enabled. The CCFn bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. If both CAPPn and CAPNn bits are set to logic 1, then the state of the Port pin associated with CEXn can be read directly to determine whether a rising-edge or falling-edge caused the capture.

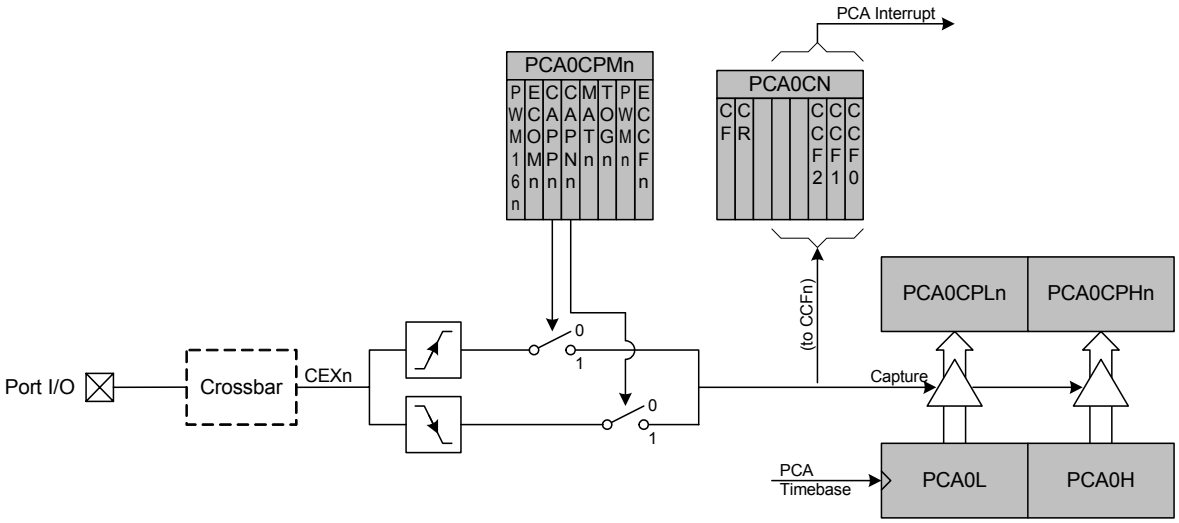


Figure 20.4. PCA Capture Mode Diagram

Note: The CEXn input signal must remain high or low for at least 2 system clock cycles to be recognized by the hardware.

20.2.2. Software Timer (Compare) Mode

In Software Timer mode, the PCA counter/timer value is compared to the module's 16-bit capture/compare register (PCA0CPHn and PCA0CPLn). When a match occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic 1 and an interrupt request is generated if CCF interrupts are enabled. The CCFn bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. Setting the ECOMn and MATn bits in the PCA0CPMn register enables Software Timer mode.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to '0'; writing to PCA0CPHn sets ECOMn to '1'.

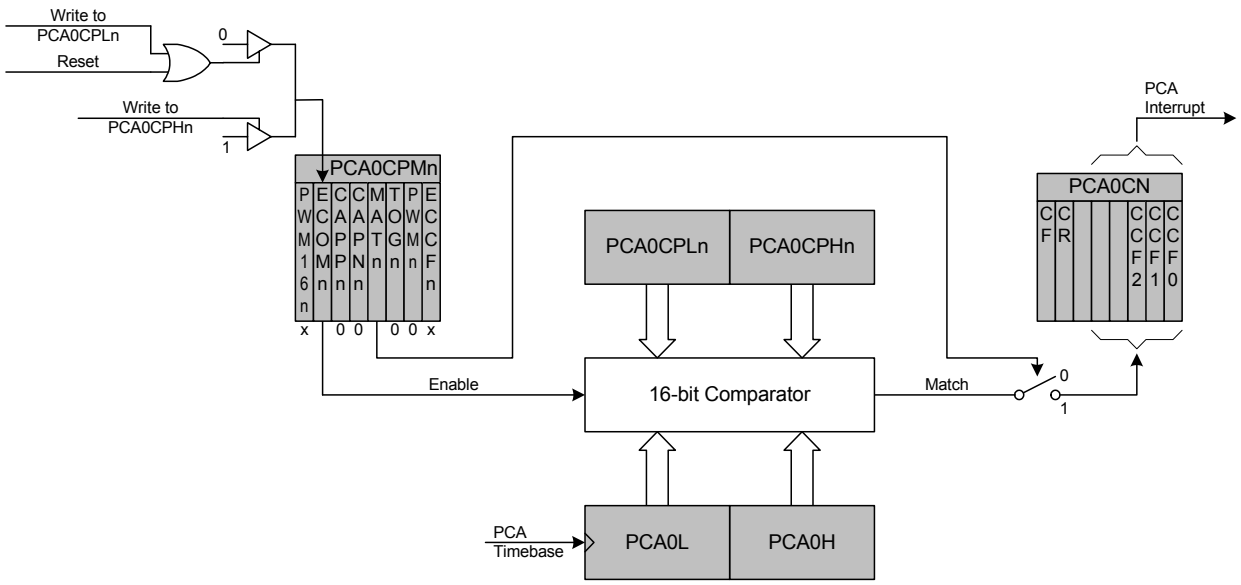


Figure 20.5. PCA Software Timer Mode Diagram

20.2.3. High Speed Output Mode

In High Speed Output mode, a module's associated CEXn pin is toggled each time a match occurs between the PCA Counter and the module's 16-bit capture/compare register (PCA0CPHn and PCA0CPLn) Setting the TOGn, MATn, and ECOMn bits in the PCA0CPMn register enables the High-Speed Output mode.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to '0'; writing to PCA0CPHn sets ECOMn to '1'.

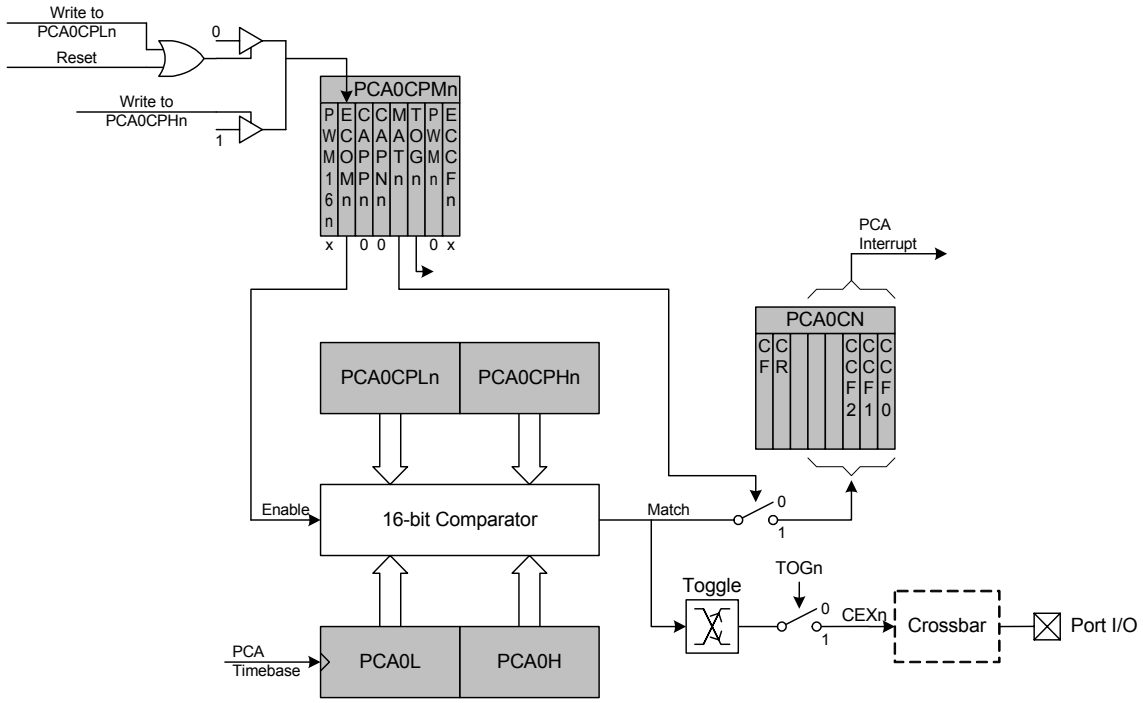


Figure 20.6. PCA High-Speed Output Mode Diagram

Note: The initial state of the Toggle output is logic 1 and is initialized to this state when the module enters High Speed Output Mode.

20.2.4. Frequency Output Mode

Frequency Output Mode produces a programmable-frequency square wave on the module's associated CEXn pin. The capture/compare module high byte holds the number of PCA clocks to count before the output is toggled. The frequency of the square wave is then defined by Equation 20.1.

$$F_{CEXn} = \frac{F_{PCA}}{2 \times PCA0CPHn}$$

Note: A value of 0x00 in the PCA0CPHn register is equal to 256 for this equation.

Equation 20.1. Square Wave Frequency Output

Where F_{PCA} is the frequency of the clock selected by the CPS2-0 bits in the PCA mode register, PCA0MD. The lower byte of the capture/compare module is compared to the PCA counter low byte; on a match, CEXn is toggled and the offset held in the high byte is added to the matched value in PCA0CPLn. Frequency Output Mode is enabled by setting the ECOMn, TOGn, and PWMn bits in the PCA0CPMn register.

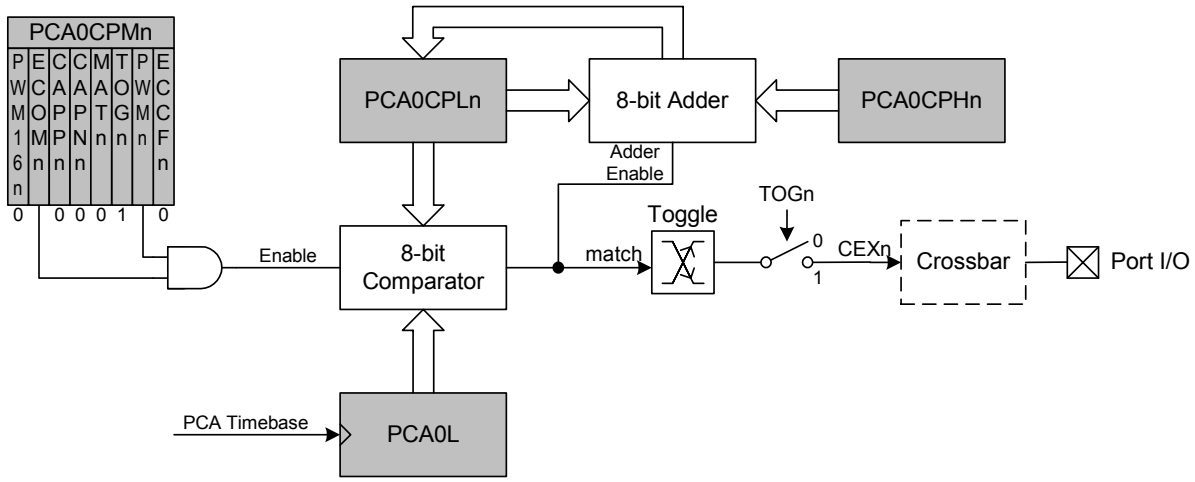


Figure 20.7. PCA Frequency Output Mode

20.2.5. 8-Bit Pulse Width Modulator Mode

Each module can be used independently to generate a pulse width modulated (PWM) output on its associated CEXn pin. The frequency of the output is dependent on the timebase for the PCA counter/timer. The duty cycle of the PWM output signal is varied using the module's PCA0CPHn capture/compare register. When the value in the low byte of the PCA counter/timer (PCA0L) is equal to the value in PCA0CPLn, the output on the CEXn pin will be set. When the count value in PCA0L overflows, the CEXn output will be reset (see Figure 20.8). Also, when the counter/timer low byte (PCA0L) overflows from 0xFF to 0x00, PCA0CPLn is reloaded automatically with the value stored in the module's capture/compare high byte (PCA0CPHn) without software intervention. Setting the ECOMn and PWMn bits in the PCA0CPMn register enables 8-Bit Pulse Width Modulator mode. The duty cycle for 8-Bit PWM Mode is given by Equation 20.2.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to '0'; writing to PCA0CPHn sets ECOMn to '1'.

$$DutyCycle = \frac{(256 - PCA0CPHn)}{256}$$

Equation 20.2. 8-Bit PWM Duty Cycle

Using Equation 20.2, the largest duty cycle is 100% (PCA0CPHn = 0), and the smallest duty cycle is 0.39% (PCA0CPHn = 0xFF). A 0% duty cycle may be generated by clearing the ECOMn bit to '0'.

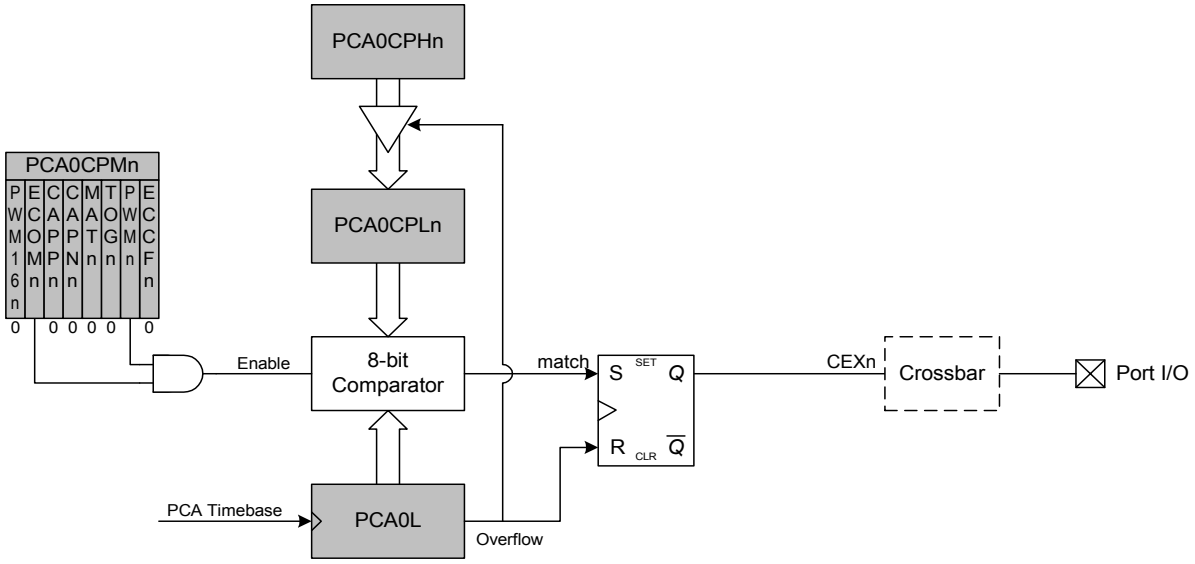


Figure 20.8. PCA 8-Bit PWM Mode Diagram

20.2.6. 16-Bit Pulse Width Modulator Mode

A PCA module may also be operated in 16-Bit PWM mode. In this mode, the 16-bit capture/compare module defines the number of PCA clocks for the low time of the PWM signal. When the PCA counter matches the module contents, the output on CEXn is asserted high; when the counter overflows, CEXn is asserted low. To output a varying duty cycle, new value writes should be synchronized with PCA CCFn match interrupts. 16-Bit PWM Mode is enabled by setting the ECOMn, PWMn, and PWM16n bits in the PCA0CPMn register. For a varying duty cycle, match interrupts should be enabled (ECCFn = 1 AND MATn = 1) to help synchronize the capture/compare register writes. The duty cycle for 16-Bit PWM Mode is given by Equation 20.3.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to '0'; writing to PCA0CPHn sets ECOMn to '1'.

$$DutyCycle = \frac{(65536 - PCA0CPn)}{65536}$$

Equation 20.3. 16-Bit PWM Duty Cycle

Using Equation 20.3, the largest duty cycle is 100% (PCA0CPn = 0), and the smallest duty cycle is 0.0015% (PCA0CPn = 0xFFFF). A 0% duty cycle may be generated by clearing the ECOMn bit to '0'.

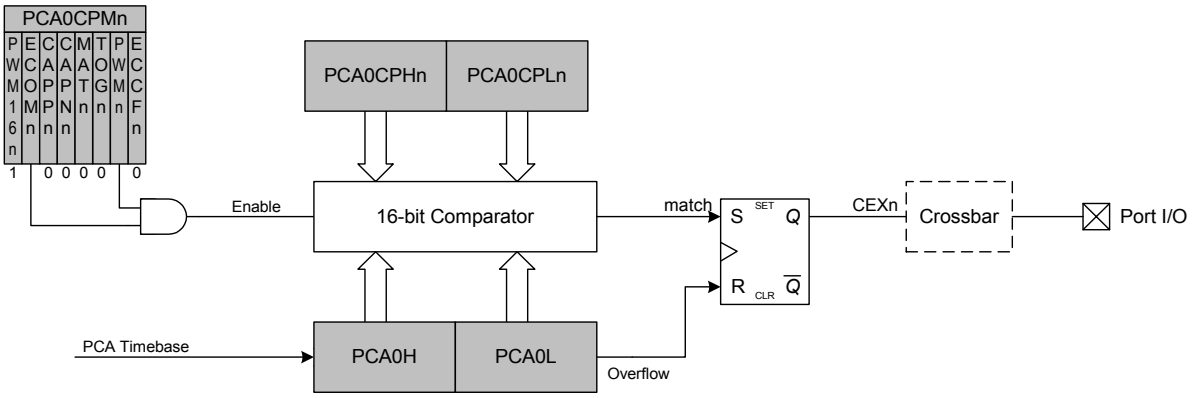


Figure 20.9. PCA 16-Bit PWM Mode

20.3. Watchdog Timer Mode

A programmable watchdog timer (WDT) function is available through the PCA Module 2. The WDT is used to generate a reset if the time between writes to the WDT update register (PCA0CPH2) exceed a specified limit. The WDT can be configured and enabled/disabled as needed by software.

With the WDTE bit set in the PCA0MD register, Module 2 operates as a watchdog timer (WDT). The Module 2 high byte is compared to the PCA counter high byte; the Module 2 low byte holds the offset to be used when WDT updates are performed. **The Watchdog Timer is enabled on reset. Writes to some PCA registers are restricted while the Watchdog Timer is enabled.**

20.3.1. Watchdog Timer Operation

While the WDT is enabled:

- PCA counter is forced on.
- Writes to PCA0L and PCA0H are not allowed.
- PCA clock source bits (CPS2-CPS0) are frozen.
- PCA Idle control bit (CIDL) is frozen.
- Module 2 is forced into software timer mode.
- Writes to the Module 2 mode register (PCA0CPM2) are disabled.

While the WDT is enabled, writes to the CR bit will not change the PCA counter state; the counter will run until the WDT is disabled. The PCA counter run control (CR) will read zero if the WDT is enabled but user software has not enabled the PCA counter. If a match occurs between PCA0CPH2 and PCA0H while the WDT is enabled, a reset will be generated. To prevent a WDT reset, the WDT may be updated with a write of any value to PCA0CPH2. Upon a PCA0CPH2 write, PCA0H plus the offset held in PCA0CPL2 is loaded into PCA0CPH2 (See Figure 20.10).

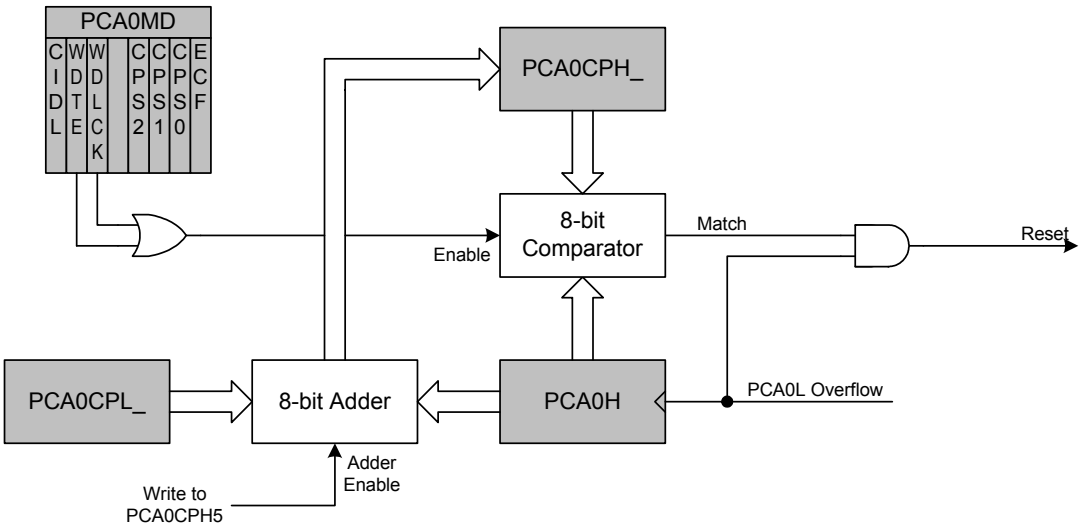


Figure 20.10. PCA Module 2 with Watchdog Timer Enabled

Note that the 8-bit offset held in PCA0CPH2 is compared to the upper byte of the 16-bit PCA counter. This offset value is the number of PCA0L overflows before a reset. Up to 256 PCA clocks may pass before the first PCA0L overflow occurs, depending on the value of the PCA0L when the update is performed. The total offset is then given (in PCA clocks) by Equation 20.4, where PCA0L is the value of the PCA0L register at the time of the update.

$$Offset = (256 \times PCA0CPL2) + (256 - PCA0L)$$

Equation 20.4. Watchdog Timer Offset in PCA Clocks

The WDT reset is generated when PCA0L overflows while there is a match between PCA0CPH2 and PCA0H. Software may force a WDT reset by writing a '1' to the CCF2 flag (PCA0CN.2) while the WDT is enabled.

20.3.2. Watchdog Timer Usage

To configure the WDT, perform the following tasks:

- Disable the WDT by writing a '0' to the WDTE bit.
- Select the desired PCA clock source (with the CPS2-CPS0 bits).
- Load PCA0CPL2 with the desired WDT update offset value.
- Configure the PCA Idle mode (set CIDL if the WDT should be suspended while the CPU is in Idle mode).
- Enable the WDT by setting the WDTE bit to '1'.

The PCA clock source and Idle mode select cannot be changed while the WDT is enabled. The watchdog timer is enabled by setting the WDTE or WDLCK bits in the PCA0MD register. When WDLCK is set, the WDT cannot be disabled until the next system reset. If WDLCK is not set, the WDT is disabled by clearing the WDTE bit.

The WDT is enabled following any reset. The PCA0 counter clock defaults to the system clock divided by 12, PCA0L defaults to 0x00, and PCA0CPL2 defaults to 0x00. Using Equation 20.4, this results in a WDT timeout interval of 3072 system clock cycles. Table 20.3 lists some example timeout intervals for typical system clocks.

Table 20.3. Watchdog Timer Timeout Intervals[1]

System Clock (Hz)	PCA0CPL2	Timeout Interval (ms)
24,500,000	255	32.1
24,500,000	128	16.2
24,500,000	32	4.1
18,432,000	255	42.7
18,432,000	128	21.5
18,432,000	32	5.5
11,059,200	255	71.1
11,059,200	128	35.8
11,059,200	32	9.2
3,062,500	255	257
3,062,500	128	129.5
3,062,500	32	33.1
191,406[2]	255	4109
191,406[2]	128	2070
191,406[2]	32	530
32,000	255	24576
32,000	128	12384
32,000	32	3168

Notes:

1. Assumes SYSCLK / 12 as the PCA clock source, and a PCA0L value of 0x00 at the update time.
2. Internal oscillator reset frequency.

20.4. Register Descriptions for PCA

Following are detailed descriptions of the special function registers related to the operation of the PCA.

SFR Definition 20.1. PCA0CN: PCA Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CF	CR	—	—	—	CCF2	CCF1	CCF0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0xD8

Bit7: **CF**: PCA Counter/Timer Overflow Flag.
Set by hardware when the PCA Counter/Timer overflows from 0xFFFF to 0x0000. When the Counter/Timer Overflow (CF) interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit6: **CR**: PCA Counter/Timer Run Control.
This bit enables/disables the PCA Counter/Timer.
0: PCA Counter/Timer disabled.
1: PCA Counter/Timer enabled.

Bits5–3: **Reserved**.

Bit2: **CCF2**: PCA Module 2 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF2 interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit1: **CCF1**: PCA Module 1 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF1 interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

Bit0: **CCF0**: PCA Module 0 Capture/Compare Flag.
This bit is set by hardware when a match or capture occurs. When the CCF0 interrupt is enabled, setting this bit causes the CPU to vector to the PCA interrupt service routine. This bit is not automatically cleared by hardware and must be cleared by software.

SFR Definition 20.2. PCA0MD: PCA Mode

R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	Reset Value
CIDL	WDTE	WDLCK	-	CPS2	CPS1	CPS0	ECF	01000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0xD9

Bit7: **CIDL**: PCA Counter/Timer Idle Control.
Specifies PCA behavior when CPU is in Idle Mode.
0: PCA continues to function normally while the system controller is in Idle Mode.
1: PCA operation is suspended while the system controller is in Idle Mode.

Bit6: **WDTE**: Watchdog Timer Enable
If this bit is set, PCA Module 2 is used as the watchdog timer.
0: Watchdog Timer disabled.
1: PCA Module 2 enabled as Watchdog Timer.

Bit5: **WDLCK**: Watchdog Timer Lock
This bit locks/unlocks the Watchdog Timer Enable. When WDLCK is set, the Watchdog Timer may not be disabled until the next system reset.
0: Watchdog Timer Enable unlocked.
1: Watchdog Timer Enable locked.

Bit4: **UNUSED**. Read = 0b, Write = don't care.

Bits3–1: **CPS2–CPS0**: PCA Counter/Timer Pulse Select.
These bits select the timebase source for the PCA counter.

CPS2	CPS1	CPS0	Timebase
0	0	0	System clock divided by 12
0	0	1	System clock divided by 4
0	1	0	Timer 0 overflow
0	1	1	High-to-low transitions on ECI (max rate = system clock divided by 4)
1	0	0	System clock
1	0	1	External clock divided by 8*
1	1	0	RTC clock divided by 8*
1	1	1	Reserved

***Note:** External clock divided by 8 and RTC0 clock divided by 8 are synchronized with the system clock.

Bit0: **ECF**: PCA Counter/Timer Overflow Interrupt Enable.
This bit sets the masking of the PCA Counter/Timer Overflow (CF) interrupt.
0: Disable the CF interrupt.
1: Enable a PCA Counter/Timer Overflow interrupt request when CF (PCA0CN.7) is set.

Note: When the WDTE bit is set to '1', the PCA0MD register cannot be modified. To change the contents of the PCA0MD register, the Watchdog Timer must first be disabled.

SFR Definition 20.3. PCA0CPMn: PCA Capture/Compare Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
PWM16n	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: PCA0CPM0: 0xDA, PCA0CPM1: 0xDB, PCA0CPM2: 0xDC

Bit7: **PWM16n**: 16-bit Pulse Width Modulation Enable.
This bit selects 16-bit mode when Pulse Width Modulation mode is enabled (PWMn = 1).
0: 8-bit PWM selected.
1: 16-bit PWM selected.

Bit6: **ECOMn**: Comparator Function Enable.
This bit enables/disables the comparator function for PCA module n.
0: Disabled.
1: Enabled.

Bit5: **CAPPn**: Capture Positive Function Enable.
This bit enables/disables the positive edge capture for PCA module n.
0: Disabled.
1: Enabled.

Bit4: **CAPNn**: Capture Negative Function Enable.
This bit enables/disables the negative edge capture for PCA module n.
0: Disabled.
1: Enabled.

Bit3: **MATn**: Match Function Enable.
This bit enables/disables the match function for PCA module n. When enabled, matches of the PCA counter with a module's capture/compare register cause the CCFn bit in PCA0MD register to be set to logic 1.
0: Disabled.
1: Enabled.

Bit2: **TOGn**: Toggle Function Enable.
This bit enables/disables the toggle function for PCA module n. When enabled, matches of the PCA counter with a module's capture/compare register cause the logic level on the CEXn pin to toggle. If the PWMn bit is also set to logic 1, the module operates in Frequency Output Mode.
0: Disabled.
1: Enabled.

Bit1: **PWMn**: Pulse Width Modulation Mode Enable.
This bit enables/disables the PWM function for PCA module n. When enabled, a pulse width modulated signal is output on the CEXn pin. 8-bit PWM is used if PWM16n is cleared; 16-bit mode is used if PWM16n is set to logic 1. If the TOGn bit is also set, the module operates in Frequency Output Mode.
0: Disabled.
1: Enabled.

Bit0: **ECCFn**: Capture/Compare Flag Interrupt Enable.
This bit sets the masking of the Capture/Compare Flag (CCFn) interrupt.
0: Disable CCFn interrupts.
1: Enable a Capture/Compare Flag interrupt request when CCFn is set.

SFR Definition 20.4. PCA0L: PCA Counter/Timer Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xF9

Bits7–0: **PCA0L**: PCA Counter/Timer Low Byte.
The PCA0L register holds the low byte (LSB) of the 16-bit PCA Counter/Timer.

SFR Definition 20.5. PCA0H: PCA Counter/Timer High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:

SFR Address: 0xFA

Bits7–0: **PCA0H**: PCA Counter/Timer High Byte.
The PCA0H register holds the high byte (MSB) of the 16-bit PCA Counter/Timer.

SFR Definition 20.6. PCA0CPLn: PCA Capture Module Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: PCA0CPL0: 0xFB, PCA0CPL1: 0xE9, PCA0CPL2: 0xEB

Bits7–0: **PCA0CPLn**: PCA Capture Module Low Byte.
The PCA0CPLn register holds the low byte (LSB) of the 16-bit capture module n.

SFR Definition 20.7. PCA0CPHn: PCA Capture Module High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: PCA0CPH0: 0xFC, PCA0CPH1: 0xE9, PCA0CPH2: 0xEC

Bits7–0: **PCA0CPHn**: PCA Capture Module High Byte.
The PCA0CPHn register holds the high byte (MSB) of the 16-bit capture module n.

21. Revision Specific Behavior

This chapter contains behavioral differences between C8051F52x/F53x "REV_A" and “REV_B" or later devices.

These differences do not affect the functionality or performance of most systems and are described below.

21.1. Revision Identification

The Lot ID Code on the top side of the device package can be used for decoding device revision information. On C8051F52x and C8051F53x devices the revision letter is the first letter of the Lot ID Code.

Figures 21.1, 21.2, and 21.3 show how to find the Lot ID Code on the top side of the device package.

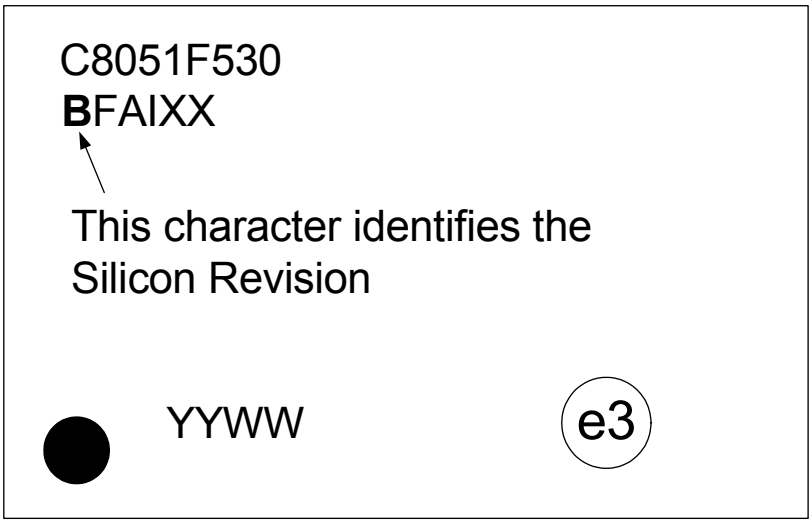


Figure 21.1. Device Package - TSSOP 20

Figure 21.2. Device Package - QFN 20

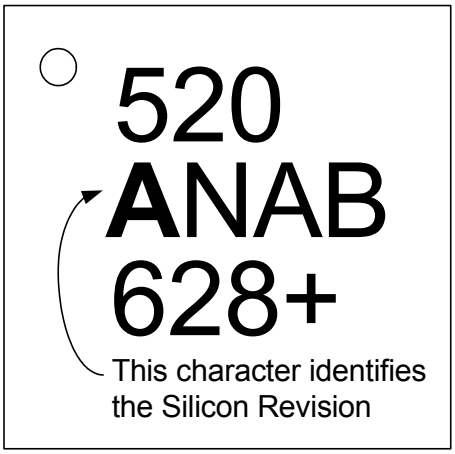


Figure 21.3. Device Package - QFN 10

21.2. Reset Behavior

The reset behavior of C8051F52x and C8051F53x "REV A" devices is different than "REV B" and later devices. The differences affect the state of the RST pin during a VDD Monitor reset.

On "REV A" devices, a V_{DD} Monitor reset does not affect the state of the RST pin. On "REV B" and later devices, a V_{DD} Monitor reset will pull the RST pin low for the duration of the brownout condition.

21.3. UART Pins

The reset behavior of C8051F52x and C8051F53x "REV A" devices is different than "REV B" and later devices. The location of the pins used by the serial UART interface is different between "REV A" and "REV B" devices.

On "REV A" devices, the TX and RX pins used by the UART interface are mapped to the P0.3 (TX) and P0.4 (RX) pins. On "REV B" and later devices, the TX and RX pins used by the UART interface are mapped to the P0.4 (TX) and P0.5 (RX) pins.

21.4. LIN

The LIN peripheral behavior in "REV A" is different than the behavior of "REV B" and later devices. The differences are:

21.4.1. Stop Bit Check

On "REV A" devices, the stop bits of the fields in the LIN frame are not checked and no error is generated if the stop bits could not be sent or received correctly. On "REV B" and later devices, the stop bits are checked, and an error will be generated if the stop bit was not sent or received correctly.

21.4.2. Synch Break and Synch Field Length Check

On "REV A" devices, the check of sync field length versus sync break length is incorrect. On "REV B" and later devices, the sync break length must be larger than 10 bit times (of the measured bit time) to enable the synchronization.

22. C2 Interface

C8051F52x/F53x devices include an on-chip Silicon Laboratories 2-Wire (C2) debug interface to allow Flash programming and in-system debugging with the production part installed in the end application. The C2 interface uses a clock signal (C2CK) and a bi-directional C2 data signal (C2D) to transfer information between the device and a host system. See the C2 Interface Specification for details on the C2 protocol.

22.1. C2 Interface Registers

The following describes the C2 registers necessary to perform Flash programming functions through the C2 interface. All C2 registers are accessed through the C2 interface as described in the C2 Interface Specification.

C2 Register Definition 22.1. C2ADD: C2 Address

								Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Bits7–0: The C2ADD register is accessed via the C2 interface to select the target Data register for C2 Data Read and Data Write commands.

Address	Description
0x00	Selects the Device ID register for Data Read instructions (DEVICEID)
0x01	Selects the Revision ID register for Data Read instructions (REVID)
0x02	Selects the C2 Flash Programming Control register for Data Read/Write instructions (FPCTL)
0xB4	Selects the C2 Flash Programming Data register for Data Read/Write instructions (FPDAT)

C2 Register Definition 22.2. DEVICEID: C2 Device ID

								Reset Value
								00010001
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

This read-only register returns the 8-bit device ID: 0x11 (C8051F52x/C8051F53x).

C2 Register Definition 22.3. REVID: C2 Revision ID

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Reset Value
								Varies

This read-only register returns the 8-bit revision ID.
For example, 0x00 = Revision A.

C2 Register Definition 22.4. FPCTL: C2 Flash Programming Control

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Reset Value
								00000000

Bits7–0 **FPCTL**: Flash Programming Control Register.
This register is used to enable Flash programming via the C2 interface. To enable C2 Flash programming, the following codes must be written in order: 0x02, 0x01. Note that once C2 Flash programming is enabled, a system reset must be issued to resume normal operation.

C2 Register Definition 22.5. FPDAT: C2 Flash Programming Data

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Reset Value
								00000000

Bits7–0: **FPDAT**: C2 Flash Programming Data Register.
This register is used to pass Flash commands, addresses, and data during C2 Flash accesses. Valid commands are listed below.

Code	Command
0x06	Flash Block Read
0x07	Flash Block Write
0x08	Flash Page Erase
0x03	Device Erase

22.2. C2 Pin Sharing

The C2 protocol allows the C2 pins to be shared with user functions so that in-system debugging and Flash programming functions may be performed. This is possible because C2 communication is typically performed when the device is in the halt state, where all on-chip peripherals and user software are stalled. In this halted state, the C2 interface can safely ‘borrow’ the C2CK (/RST) and C2D (P2.7) pins. In most applications, external resistors are required to isolate C2 interface traffic from the user application. A typical isolation configuration is shown in Figure 22.1.

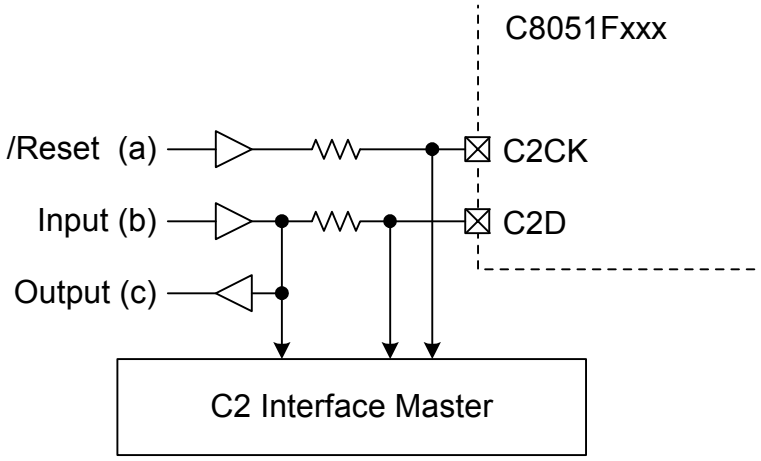


Figure 22.1. Typical C2 Pin Sharing

The configuration in Figure 22.1 assumes the following:

1. The user input (b) cannot change state while the target device is halted.
2. The /RST pin on the target device is used as an input only.

Additional resistors may be necessary depending on the specific application.

CONTACT INFORMATION

Silicon Laboratories Inc.

400 West Cesar Chavez
Austin, TX 78701
Tel: 1+(512) 416-8500
Fax: 1+(512) 416-9669
Toll Free: 1+(877) 444-3032

Email: MCUinfo@silabs.com
Internet: www.silabs.com